JMobile Suite User Manual v1.91

This document contains information for JMobile Suite on-line help, accessible from JMobile Studio \Help command

Exor International S.p.A.
ptn0461-028.doc

Copyright © 2012 Exor International S.p.A.

Subject to change without notice
The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind.
Third-party brands and names are the property of their respective owners.
www.exorint.net

Microsoft®, Win32, Windows®, Windows XP, Windows Vista, Windows 7, Windows 8, Visual Studio are either registered trademarks or trademarks of the Microsoft Corporation in the United States and other countries. Other products and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logo, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred.

Contents

Contents 3

1 Getting Started 12
- 1.1 Assumptions 12
- 1.2 Installing the Software 12
- 1.2.1 System Requirements 12
- 1.2.2 Installation 13
- 1.2.3 Licensing 14
- 1.2.4 Installing Multiple Versions of JMobile Suite on the Same Computer 15
- 1.2.5 Opening Projects Created with Older Version of JMobile Studio 16
- 1.2.6 Multilanguage for JMobile Studio 17
- 1.2.7 Crash report 17
- 1.3 Installing Runtime PC 17
- 1.3.1 System Requirements 18
- 1.3.2 Installation 18
- 1.3.3 Licensing 18
- 1.3.4 Limitations 19
- 1.3.5 Fullscreen Mode 20
- 1.3.6 WORKSPACE 20
- 1.3.7 FAQ: Why am I not able to download project to Runtime PC? 20

2 The HMI Runtime 24
- 2.1 Runtime Modes 24
- 2.2 Basic Unit Settings 24
- 2.3 Other Context Menu Options 26
- 2.4 Built-in SNTP Service 28

3 My First Project 29
- 3.1 Creating a New Project 29
- 3.2 The Workspace 31
- 3.3 Communication Protocols 33
- 3.4 Tags 34
- 3.4.1 Tag Editor 36
- 3.4.2 Data Types 37
- 3.4.3 Dictionaries 38
- 3.5 Designing a Page 39
- 3.5.1 Importing a Page 40
- 3.5.2 Dialog Pages 41
- 3.6 The Widget Gallery 42
- 3.7 “Attach To” and Dynamic Properties 45

4 The HMI Simulator 47
- 4.1 Launching the Simulator 47
- 4.2 Stopping the Simulator 47
- 4.3 Simulator Settings 47

5 Transferring the Project to Target 50
- 5.1 Download to Target 50
- 5.2 Update Package 53
- 5.3 The Runtime Loader 55
- 5.4 Upload Projects 57

6 Programming Concepts 59
- 6.1 Attach to 59

6.2 Events ... 62
6.2.1 OnClick / OnMouseClick ... 63
6.2.2 OnHold / OnMouseHold ... 63
6.2.3 Autorepeat ... 64
6.2.4 OnWheel ... 65
6.2.5 OnActivate ... 65
6.2.6 OnDataUpdate ... 65
6.3 Widgets positioning: Snap to Grid / Snap to Objects ... 65
6.3.1 Snap to Grid ... 66
6.3.2 Snap to Object ... 66
6.4 Z-order of widgets ... 66
6.5 Change properties of several widgets at once ... 67

7 Project Properties / Project Widget ... 68
7.1 Version ... 68
7.2 Context Menu ... 69
7.3 Developer Tools ... 69
7.3.1 Profiling ... 71
7.3.2 Watchdog ... 72
7.4 Buzzer on touch / Buzzer duration ... 72
7.5 Keyboard ... 72
7.6 JavaScript Debug ... 73
7.7 Allow JS Remote Debugger ... 73
7.8 Image DB enable ... 73
7.9 FreeType Font Rendering ... 73
7.10 Software Plug-in Modules ... 73
7.11 Behaviour -> Home Page ... 74
7.11.1 Behavior -> Page Width / Page Height ... 74
7.11.2 Behavior -> Display Mode ... 74
7.11.3 Behavior -> Project Type ... 74
7.11.4 Behavior -> PageRequest, CurrentPage and SyncOptions ... 74
7.11.5 Behavior -> Hold Time / Autorepeat Time ... 78
7.11.6 Events -> OnWheel ... 78

8 System Variables ... 79
8.1 Alarms ... 79
8.2 Communication ... 80
8.3 Daylight Saving Time ... 80
8.4 Device ... 81
8.5 Dump Information ... 83
8.6 Keypad ... 83
8.7 Network ... 83
8.8 PLC ... 84
8.9 Printing ... 84
8.10 Screen ... 85
8.11 SD Card ... 85
8.12 Server ... 85
8.13 Time ... 86
8.14 USB Drive ... 86
8.15 User Management ... 86
8.16 Version ... 87

9 Actions ... 88
9.1 Widget Actions ... 88
9.1.1 ShowWidget ... 88
9.1.2 TriggerIPCamera ... 89
9.1.3 SlideWidget ... 89
9.1.4 RefreshEvent ... 91
9.1.5 ContextMenu ... 91

9.1.6 ReplaceMedia ... 92
9.2 Media Actions ... 93
9.2.1 PlayMedia ... 93
9.2.2 StopMedia ... 93
9.2.3 ReloadMedia ... 93
9.2.4 PauseMedia ... 93
9.2.5 BrowseMedia ... 93
9.3 Keyboard Actions ... 94
9.3.1 SendKey ... 94
9.3.2 SendKeyWidget ... 95
9.3.3 ShowKeyPad ... 97
9.3.4 KeyboardMacros ... 97
9.4 Page Actions ... 98
9.4.1 LoadPage ... 98
9.4.2 HomePage ... 99
9.4.3 PrevPage ... 99
9.4.4 NextPage ... 99
9.4.5 LastVisitedPage ... 99
9.4.6 ShowDialog ... 99
9.4.7 CloseDialog ... 100
9.4.8 ShowMessage ... 100
9.4.9 LaunchApplication ... 101
9.4.10 LaunchBrowser ... 101
9.4.11 LaunchVNC ... 101
9.4.12 LaunchPDFViewer ... 101
9.4.13 LaunchUpdater ... 102
9.4.14 LockScreen ... 102
9.5 MultiLang Actions ... 102
9.5.1 SetLanguage ... 102
9.6 Tag Actions ... 102
9.6.1 DataTransfer ... 102
9.6.2 ToggleBit ... 102
9.6.3 SetBit ... 103
9.6.4 ResetBit ... 103
9.6.5 WriteTag ... 103
9.6.6 StepTag ... 103
9.6.7 ActivateGroup ... 103
9.6.8 DeactivateGroup ... 103
9.7 Trend Actions ... 103
9.7.1 RefreshTrend ... 104
9.7.2 Scroll Left Trend ... 104
9.7.3 Scroll Right Trend ... 104
9.7.4 PageLeftTrend ... 104
9.7.5 Page Right Trend ... 104
9.7.6 Page Duration Trend ... 104
9.7.7 Zoom In Trend ... 104
9.7.8 ZoomOutTrend ... 104
9.7.9 Zoom Reset Trend ... 105
9.7.10 Pause Trend ... 105
9.7.11 ResumeTrend ... 105
9.7.12 Show Trend Cursor ... 105
9.7.13 ScrollTrendCursor ... 106
9.7.14 ScrollTrendtoTime ... 106
9.7.15 ConsumptionMeterPageScroll ... 107
9.8 Alarm Actions ... 108
9.8.1 SelectAllAlarms ... 109
9.8.2 AckAlarm ... 109
9.8.3 ResetAlarm ... 109

9.8.4 EnableAlarms ... 109
9.9 Event Actions ... 110
9.9.1 ScrollEventsBackward ... 110
9.9.2 ScrollEventsForward ... 110
9.10 System Actions ... 110
9.10.1 Restart ... 110
9.10.2 ControlUserLED ... 111
9.10.3 DumpTrend ... 111
9.10.4 DeleteTrend ... 114
9.10.5 DumpEventArchive ... 115
9.10.6 DeleteEventArchive ... 116
9.10.7 ResetProtoErrCount ... 116
9.10.8 SafelyRemoveMedia ... 117
9.10.9 CopyCodesysProject ... 117
9.11 Recipe Actions ... 118
9.11.1 DownLoadRecipe ... 118
9.11.2 UpLoadRecipe ... 118
9.11.3 WriteCurrentRecipeSet ... 119
9.11.4 DownLoadCurRecipe ... 120
9.11.5 UploadCurRecipe ... 121
9.11.6 ResetRecipe ... 122
9.11.7 DumpRecipeData ... 123
9.11.8 RestoreRecipeData ... 124
9.12 User Management Actions ... 125
9.12.1 LogOut ... 125
9.12.2 SwitchUser ... 126
9.12.3 ResetPassword ... 128
9.12.4 AddUser ... 129
9.12.5 DeleteUser ... 130
9.12.6 EditUsers ... 131
9.12.7 DeleteUMDynamicFile ... 132
9.12.8 ExportUsers ... 132
9.12.9 ImportUsers ... 133
9.13 Print Actions ... 135
9.13.1 PrintGraphicReport ... 135
9.13.2 PrintText ... 135
9.13.3 EmptyPrintQueue ... 135
9.13.4 PausePrinting ... 136
9.13.5 ResumePrinting ... 136
9.13.6 AbortPrinting ... 136

10 Using JMobile Client ... 137
10.1 The JMobile Client toolbar ... 137
10.2 Settings &Time Zone Options ... 137
10.3 Workspace ... 138

11 Using the Integrated FTP Server ... 139

12 Using ActiveX Client for Internet Explorer ... 140
12.1 Installing ActiveX ... 140
12.2 HTTP Access to ActiveX files ... 140
12.3 Internet Explorer Settings ... 140
12.4 Security Setting for Trusted Site Zone ... 141
12.5 Install Active X in Internet Explorer ... 143
12.6 Uninstalling Active X ... 144
12.7 ActiveX information ... 144

13 Using VNC for Remote Access ... 145
13.1 VNC Server ... 145
13.2 VNC Viewer ... 146

14 Alarms 147
14.1 Alarm Configuration Editor. 147
14.2 Alarms' State Machine 150
14.3 Setting Events 151
14.3.1 Log Events 151
14.3.2 Notify 153
14.3.3 Actions 153
14.4 Active Alarms Widget 154
14.4.1 Filters 155
14.4.2 Sorting 156
14.5 Alarms History Widget 157
14.6 Managing alarms at Runtime 158
14.7 Enable/Disable Alarms at Runtime 158
14.8 Live Data in Alarms Widget 159
14.9 Exporting Alarm Buffers as CSV file 160

15 Recipes 161
15.1 Recipe Configuration Editor 161
15.2 Configuring Recipe Sets on the Page 163
15.3 Defining Recipe Fields 163
15.4 Recipe Status 164
15.5 Configuring Recipe Widget for Runtime Execution 165
15.6 Configure Recipe Transfer Macros 165
15.7 Upload or Download Recipes during Runtime 166
15.7.1 Recipe Download through Recipe Widget in Runtime 166
15.7.2 Recipe Download or Upload through Recipe Transfer Macro in Runtime 166
15.7.3 Backup and Restore of Recipes Data 167

16 Trends 168
16.1 Real-Time Trend 168
16.2 History Trend 169
16.2.1 Trend Editor 170
16.2.2 Configuring Trend Window for History Trends 171
16.3 Trend Window Properties 173
16.3.1 Request Samples (Advanced View) 173
16.3.2 Color Bands 173
16.4 Trend Cursor 174
16.5 Exporting Trend Buffer Data to CSV file 176

17 Scatter Diagram / XY Graph 177

18 Data Transfers 178
18.1 The Data Transfer Editor 178
18.2 Data Transfer Toolbar Buttons 178
18.3 Data Transfer Fields 179
18.4 Exporting Data to .csv Files 180
18.5 Data Transfer Limitations and Recommendations 180

19 Offline Node Management 182
19.1 Offline Node Management Process 182
19.2 Manual Offline Node Management Process 183
19.3 Manual Offline Configuration 183
19.4 Automatic Offline NodeDetection 185
19.5 Offline Management Toolbar buttons 186
19.6 Offline Management Fields 186

20 Multi-Language 187
20.1 Add a Language to Project 188
20.1.1 Language Display Combo 189
20.2 Multi-Language Widget 189
20.2.1 Multi-Language for Static Text Widget 189

20.2.2 Multi-Language for Message Widget ... 190
20.2.3 Multi-Language for Alarm Messages ... 191
20.2.4 Multi-Language for Pop-up Messages ... 191
20.3 Export and Import of Multi-language Strings ... 192
20.4 Change Languages at Runtime ... 194
20.5 Limitations in UNICODE support ... 195

21 Scheduler ... 196
21.1 Configuring the Scheduler Engine ... 196
21.2 HighResolution ... 197
21.3 Recurrence Scheduler ... 198
21.4 Configuring Location in JMobile Studio ... 199
21.5 Configuring the Schedule Interface for Runtime Interaction ... 201
21.6 Schedule the Events during Runtime ... 201

22 User Management and Passwords ... 203
22.1 Configuring Security Options ... 204
22.2 Configuring Groups and Authorizations ... 204
22.3 Modifying the Access Permission of Groups ... 204
22.3.1 Widget Permissions ... 205
22.3.2 Action Permissions ... 207
22.3.3 FTP Authorizations ... 207
22.3.4 HTTP Authorizations ... 207
22.3.5 Miscellaneous ... 208
22.3.6 Access Priority ... 209
22.4 Configuring Users ... 210
22.5 Default User ... 211
22.6 Assigning Widget Permissions from Page View ... 211
22.7 Operation on Runtime ... 212
22.8 Force Remote Login ... 214

23 Audit Trails ... 215
23.1 Enable or Disable the Audit Trail ... 215
23.2 Configure Audit Events ... 216
23.3 Configure Tags in the Audit Trail ... 216
23.4 Configure Alarms in the Audit Trail ... 217
23.5 Configure Login or Logout Details in Audit Trail. ... 218
23.6 Viewing Audit Trails in Runtime ... 219
23.7 Exporting Audit Trail as CSV File ... 219

24 Reports ... 220
24.1 Adding a report ... 220
24.2 Text Report ... 220
24.3 Graphic Report ... 221
24.3.1 Page body ... 222
24.3.2 Header and Footer ... 222
24.3.3 The Context Widget Gallery ... 222
24.3.4 Printer Configuration ... 222
24.3.5 Supported Printers ... 222
24.3.6 Printer tested ... 223
24.4 Print Events ... 224
24.5 Minimum requirements ... 225

25 Screen saver ... 226

26 Backup/Restore ... 228

27 Keypads ... 230
27.1 Creating and Using Custom Keypads ... 231
27.2 Deleting or Renaming Custom Keypads ... 234
27.3 Keypad Type ... 234

27.4 Keypad Position ... 235
28 External keyboards ... 236
28.1 Search and Filter ... 237
28.2 Shows ... 238
28.3 Clear Actions ... 238
28.4 Keyboard Layout ... 239
28.5 Enable Keyboard ... 239
28.6 Configure Macro Actions for Keys ... 240
29 Tag Cross Reference ... 241
29.1 Accessing Tag Cross Reference ... 241
29.2 Using Tag Cross Reference ... 241
29.3 Tag Cross Reference: data update ... 242
29.4 Export data in csv ... 242
30 Indexed Addressing ... 243
30.1 Creating an Indexed Addressing Set ... 243
30.1.1 Autofill tag names ... 245
30.2 Using Indexed Addressing mode in pages ... 245
31 Special Widgets ... 247
31.1 Date Time Widget ... 247
31.2 RSS Feed Widget ... 248
31.3 Control List Widget ... 249
31.3.1 State ... 250
31.3.2 Selection ... 251
31.3.3 Write on Select ... 252
31.3.4 Write on Enter ... 252
31.3.5 Read Only ... 252
31.4 Variables Widget ... 252
31.4.1 Using Variables in JavaScript ... 254
31.5 Media Player Widget ... 255
How to convert a video using SUPER © video converter (by eRightSoft) ... 256
31.5.1 Using Media Player in JavaScript ... 257
31.6 IPCamera Widget ... 258
31.6.1 IPCamera tested ... 259
31.7 PTZ Controls ... 259
31.8 Multistate Image Widget ... 260
31.9 Multistate Image Multilayer ... 261
31.10 Combo Box Widget ... 263
31.11 Consumption Meter Widget ... 264
31.12 Browser widget ... 267
32 Custom Widgets ... 269
32.1 Creating a Custom Widget ... 269
32.2 Adding the Properties ... 270
32.3 Editing Custom Properties ... 273
33 Send an E-mail Message ... 274
33.1 Configure E-mail Server ... 274
33.2 Configure E-mails ... 275
34 JavaScript ... 277
34.1 Execution ... 277
34.2 Events ... 277
34.2.1 Widget Events ... 278
34.2.2 Page Events ... 279
34.2.3 System Events ... 280
34.3 Objects ... 283
34.3.1 Widget ... 283

34.3.2 Page 287
34.3.3 Group 289
34.3.4 Project 290
34.3.5 State 298
34.4 Keywords 299
34.5 Global Functions 299
34.6 Limitations 300
34.7 Debugging of JavaScript 300
34.7.1 Remote JavaScript Debugger 301

35 JM4Web HTML5-based Access 303
35.1 Supported Browsers/Platforms 303
35.1.1 PC 303
35.1.2 Tablets / Phones 303
35.2 Generating JM4Web Pages 304
35.2.1 Exporting Pages 304
35.2.2 Select Pages to Export 304
35.2.3 Setting Home Page 305
35.2.4 Start JM4Web Export 306
35.3 Testing the JM4Web Project 307
35.3.1 Using the Online Simulator 307
35.4 Downloading the JM4Web Project 308
35.4.1 Running JM4Web from a Browser 309
35.5 JM4Web Connectivity Notes 310
35.5.1 Server Disconnect 310
35.5.2 Inactivity Timeout 311
35.5.3 User Session Termination 311
35.5.4 Non-Active JM4Web Project 312
35.6 JM4Web Limitations 312
35.6.1 Widgets 312
35.6.2 Actions 313
35.6.3 Tag Attachment 313
35.6.4 XForms 313
35.6.5 Widget Event 314
35.6.6 Page Events 314
35.6.7 JavaScript API 314
35.6.8 Keypads 314
35.6.9 Dialog Page 315
35.6.10 User Management 315
35.6.11 Maximum Concurrent User Connections 315
35.7 Troubleshooting and FAQ 315
35.7.1 Enable JavaScript 315
35.7.2 Browser Cache 316
35.7.3 Using a Proxy 316
35.7.4 Why I'm not able to see changes in web pages? 316

36 System Settings Tool 317
36.1 User Mode 317
36.2 System Mode 318

37 License Activation of HMI Panel Software Modules 321
37.1 Activate Panel 321
37.2 Save License 322
37.3 Import license 324

38 Updating System Components in HMI Panels 325
38.1 List of Upgradable Components 325
38.2 Update of System Components from JMobile Studio 326
38.3 Update of the System Components via USB Flash Drive 328

39 Access Protection to HMI Devices 330
39.1 Ports & Firewalling 331
40 Factory Restore 333
41 Tips and tricks to improve performance 334
41.1 Static Optimization 334
41.1.1 Best practices for max performance 335
41.1.2 FAQ – Static optimization 337
41.1.3 Templates 337
41.2 Page caching 337
41.3 Image DB 337
41.3.1 Best Practice to use the Image DB 338
41.4 Project level 338
41.5 Runtime level 338
41.6 OS level 339
41.7 Precache 339
41.7.1 Best Practice to use the Precache 339
41.7.2 Frequently asked questions – Precache 339
42 FAQ 340
42.1 How to change fill color property according to Tag values 340
42.1.1 Using ColorPaletteCustom Xform 340
42.1.2 Connecting Color property to a String type Tag 340
43 Functional Specifications and Compatibility 342
43.1 Table of Functions and Limits 342
43.2 Compatibility 342

1 Getting Started

The JMobile Studio is a software application used to create graphical HMI pages. The JMobile Studio has a drag-and-drop interface that makes it easy to create complex display pages. The same features found in many popular Windows applications are also available in the JMobile Studio.

This document describes how to use the JMobile Studio application, and is divided into chapters that represent the key operations of the JMobile Studio. Each chapter is presented in a standalone manner, allowing you to jump from chapter to chapter, depending on the task you wish to perform.

1.1 Assumptions

We assume that those reading this manual are using the JMobile Suite software to design control panel applications that run on UniOP panels, Series 400/500/600 and on PC.
We also assume that you have a basic understanding of PCs, Microsoft Windows, and the type of network environment in which you will run the application.

1.2 Installing the Software

The JMobile Suite contains the following as part of the installation:

JMobile Studio
JMobile Studio is an application for designing custom HMI projects in a user-friendly manner, along with a variety of options in its built-in library, the Widget Gallery.

JMobile Client
JMobile Client is a light-weight application that can be used on Windows computers to remotely view and manage an application running on an HMI Runtime.

HMI Runtime
The HMI Runtime is a standalone application that runs on the UniOP HMI panels. The HMI Runtime can be installed via JMobile Studio and is design for working with WCE 6.0 OS.

Runtime PC
The Runtime PC is a standalone application that runs on Win32 platform. It is like a HMI Runtime but has been designed to work on Win32 platforms like Windows 7, Windows XP etc.

1.2.1 System Requirements

JMobile Studio has the following system requirements:

Operating System	Windows XP (SP2 or SP3) Windows Vista (SP1 or SP2) Windows 7 Windows 8
Storage	500 MB Min

RAM	512Mb
Other	One Ethernet connection

1.2.2 Installation

Insert the CD-ROM into the CD-ROM drive. If your system has Auto run enabled, the JMobile Suite installation will start automatically; otherwise run the JMobile Studio setup application.

1. Click on the **Start** button and select **Run** from the popup menu.
2. Type `D:\setup` in the box if you are running the installation from the CD drive (if your CD-ROM drive is not drive D, replace with the appropriate letter).
3. Click **Next**.

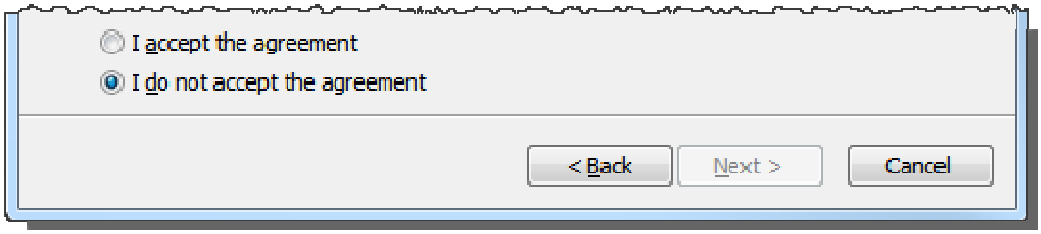


Figure 1

4. Read the JMobile Studio Software License and accept the agreement.
5. Follow the instructions on the screen. The default location for the JMobile Studio software is *C:\Program Files\Exor\JMobile Suite*. Default installation path can be changed depending on needs.
6. If the Select Components step is available, select the components you want to install.
7. The installation procedure will create a program group called *JMobile Suite* in the Start menu. A JMobile Studio icon can be added to the desktop.

Figure 2

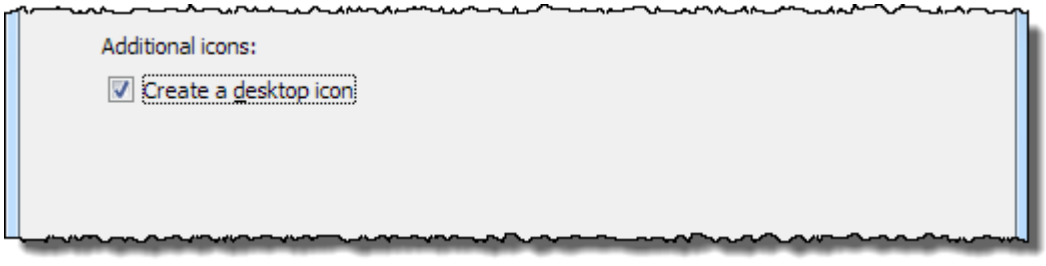


Figure 3

After installing the JMobile Studio, you can run the application by using the desktop icon or from *Start > All programs > JMobile Studio*.

1.2.3 Licensing

JMobile Studio is available with a friendly 30 days free trial policy. After 30 days since the installation the system will present a registration form where you can enter license activation keys.

Message shown after trial has expired and no valid licenses are found is: **Invalid license: license not found**. To register software before trial period expires, goto **Help** -> **Register**.

NOTE *The registration process requires an Internet connection. Ports TCP 80 and 443 are used for activation.*

When JMobile Studio performs the registration, a license file is downloaded from the License Server to the PC. License files are saved in following folders depending on OS:

- C:\Documents and Settings\[username]\Application Data\ Exor\
- C:\Users\[username]\AppData\Roaming\Exor\

Licenses are locked to the **BIOS ID** or to the **Windows product ID** of the PC where JMobile Studio is installed

If JMobile Studio is not able to reach the server (for example because no Internet connection is available), a new button appears; it can be used to activate the license via email. This mail can also be shared with technical support for assistance.

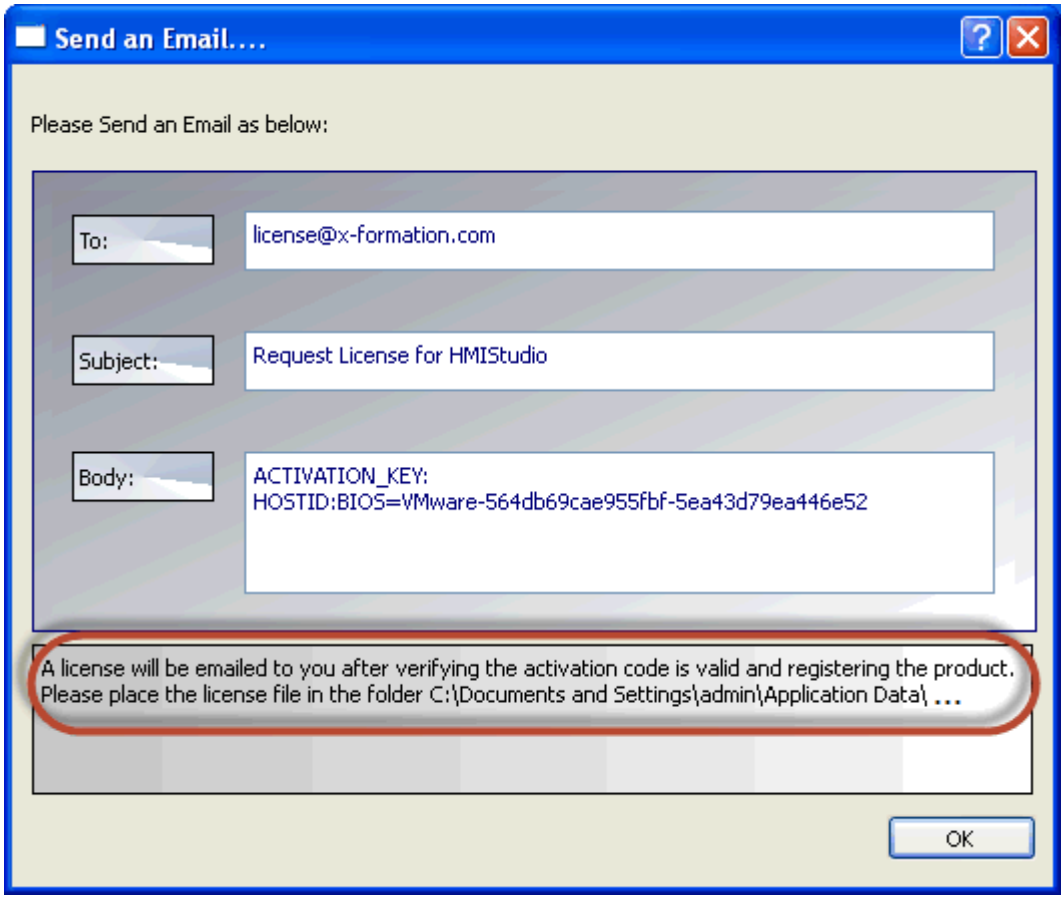

Send an Email....
Please Send an Email as below:
To:
license@x-formation.com
Subject:
Request License for HMIStudio
Body:
ACTIVATION_KEY:
HOSTID:BIOS=VMware-564db69cae955fbf-5ea43d79ea446e52
A license will be emailed to you after verifying the activation code is valid and registering the product.
Please place the license file in the folder C:\Documents and Settings\admin\Application Data\ ...
OK

Figure 4

To verify the status of an activation (in case of problems) is possible to introduce the Activation Key in the following URL: https://license.x-formation.com/licenses
From this website is possibile also to download the license file once activated and later save it in one of the following folders (based on OS):

- C:\Documents and Settings\[username]\Application Data\ Exor\
- C:\Users\[username]\AppData\Roaming\Exor\

NOTE *Trial is not supported in virtual machines. However, with a valid license virtual machines are supported.*

1.2.4 Installing Multiple Versions of JMobile Suite on the Same Computer

You may install different instances of JMobile Studio on the same computer. Each installation has its own settings and can be uninstalled individually.

During the installation process you may encounter three scenarios:

1. **First installation of JMobile Studio in the system**
   The installation procedure asks for the destination folder and installs that software in the specified folder.

2. **System with only one instance of JMobile Studio already installed**
   The installer detects that one version is already available and it will ask if you want to replace the current version with the new one or if you want to install another instance.

3. **System with multiple instances of JMobile Studio already installed**
   The installer detects that one or more installations of JMobile Studio are present and it will ask if you want to replace the **last installed** instance with the new one or if you want to install another instance (Figure 5).

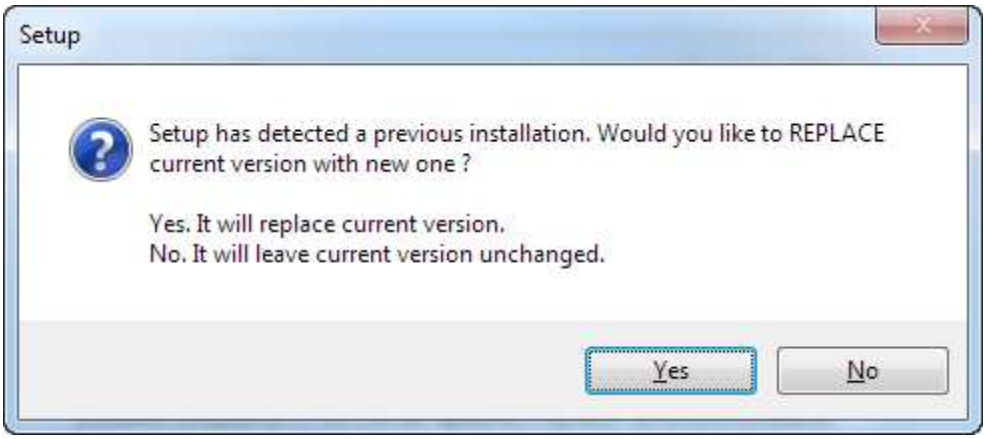

Setup
Setup has detected a previous installation. Would you like to REPLACE current version with new one ?
Yes. It will replace current version.
No. It will leave current version unchanged.
Yes
No

Figure 5

When the user tries to install a second instance of the same version of JMobile Studio, the setup procedure will detect it and will show a warning message that the same version is already available in the computer. Setup will also provide the user with the reinstall option (Figure 6).

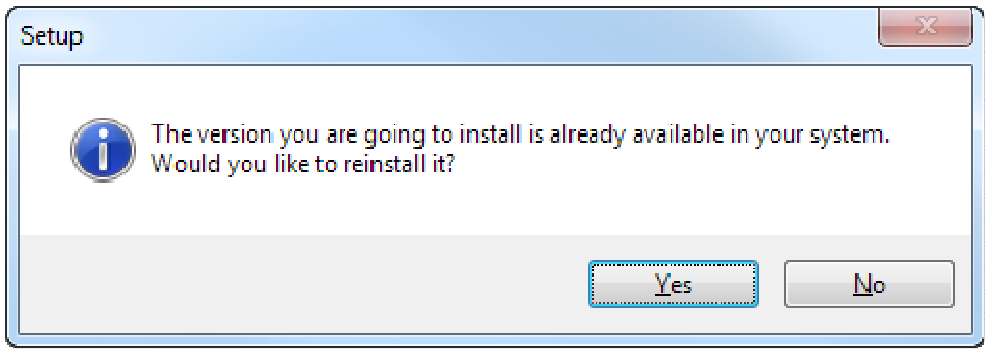


Figure 6

Each installation goes to a sub-folder which has the version number as part of its name; for instance:

C:\Program Files\Exor\JMobile Suite 1.90.

Each installed version provides its ID in the Control Panel. Add/remove program facility is provided in order to remove it individually from the computer.
Each installation identifies itself in the *Start Menu* with a program group name.

NOTE *Multiple Studio installations provide a common workspace folder for all instances of the JMobile Studio that have been installed.*

1.2.5 Opening Projects Created with Older Version of JMobile Studio

When a JMobile Studio project (file with `.jpr` extension) is opened, JMobile Studio checks for the match between the version ID stored in the jpr file and its version ID; if they match, the project will be opened normally; if they do not match, JMobile Studio shows a warning message to inform that the project has been created with a different version of JMobile Studio and report this version ID if it is available in jpr.

In this case JMobile Studio will offer two options to convert the project:

- Convert and open the project from current path. The project will be converted without a backup copy of the original version.
- Convert and save the project to a new location and Open. The older version is maintained as a backup copy.

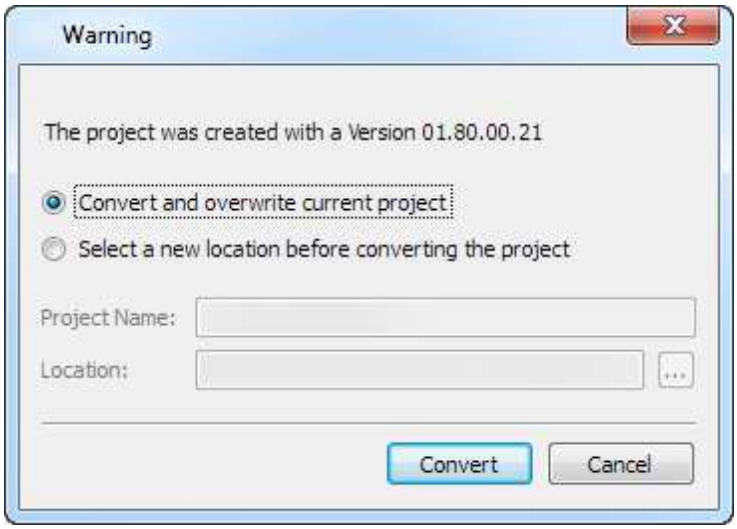


Figure 7

WARNING *Do not edit projects with a version of JMobile Studio older than the one used to create them. It can result in a damage of the project and to runtime instability.*

1.2.6 Multilanguage for JMobile Studio

Starting from v1.91 JMobile Studio is available in Multilanguage. All languages are installed by default as part of JMobile Suite.

To change default language (english) used by JMobile Studio go to *Help* -> ***Change Language*** and change current with preferred language.

1.2.7 Crash report

Crash report dialog appears automatically in case JMobile Studio freeze or crash and allows users to save a log file of crash. The crash report may contain information important for technical support.

NOTE *Crash reports are disabled in Windows XP OS*

1.3 Installing Runtime PC

Runtime PC for Windows is an unique HMI platform that combines advanced HMI features and vector graphics with powerful web technologies.You can choose the platform to monitor and control your equipment with tags, alarms, schedulers, recipes, trends, Javascript logic and events while at the same time gain ubiquitous web access, e-mail alerts and mobile display. The Runtime architecture is built from the core using Internet standards.
Runtime PC provides connectivity with the most popular factory and building automation protocols, based on Ethernet and serial interfaces. Multiple communication feature let you connect with up to 4 devices at the same time. The number of available communication protocols is continuously growing to ensure the Runtime PC connectivity with other devices.

1.3.1 System Requirements

The minimum system requirements required to use Runtime PC are the following:

Operating System	Windows XP Professional SP2/SP3 Windows XP Embedded (SP3) Windows Vista Business/Ultimate (32 bit) Windows 7 Professional (32/64 bit) Windows 7 Embedded Windows 8 Windows 2003 Server
Storage	256 MB Min
RAM	512Mb
CPU	min. 300 MHz Pentium III or similar processors with 500 MHz.
Graphic	min. SVGA min 1024x768
Other	One Ethernet connection

1.3.2 Installation

Runtime PC is distribuited as a component of the JMobile Suite. During software installation verify in Select Components that Runtime PC is checked to install it into the target system.

As described in chapter **Remote Access / Firewalling**, Runtime PC as a standard HMI Runtime requires that ports like 80/HTTP and 21/FTP are not blocked by the firewall. Runtime is able to detect if ports are in use and when a conflict is present it will show a dialog that allow user to change the default ports.

NOTE *Sometime Runtime PC is not able to detect all services running in ports like 80/HTTP and 21/FTP and this force Runtime PC to be closed automatically. This could happen for example when IIS (Internet Information Services) or MS SQL Server or other windows services are running on these ports creating the conflict. In these cases, disable window services to make it run correctly.*

1.3.3 Licensing

Runtime PC is available with a friendly 30 days free trial policy. 30 days after the installation the system will present a registration form where you can enter a license activation key.

To register Runtime PC before the trial period expires, use a right mouse click to go to the context menu and select **Register**.

NOTE *The registration process requires an Internet connection. Ports TCP 80 and 443 are used for activation.*

When Runtime PC is registered a license file is downloaded from the License Server to the PC. License files are saved in following folders depending on OS:

- C:\Documents and Settings\[username]\Application Data\ Exor
- C:\Users\[username]\AppData\Roaming\Exor

Licenses are locked to the **BIOS ID** or to the **Windows product ID** of the PC where it is installed

If Runtime PC is not able to reach server (for example because no internet connection is available), a new button appears; it can be used to activate license via email. This mail can be also shared with technical support for assistance.

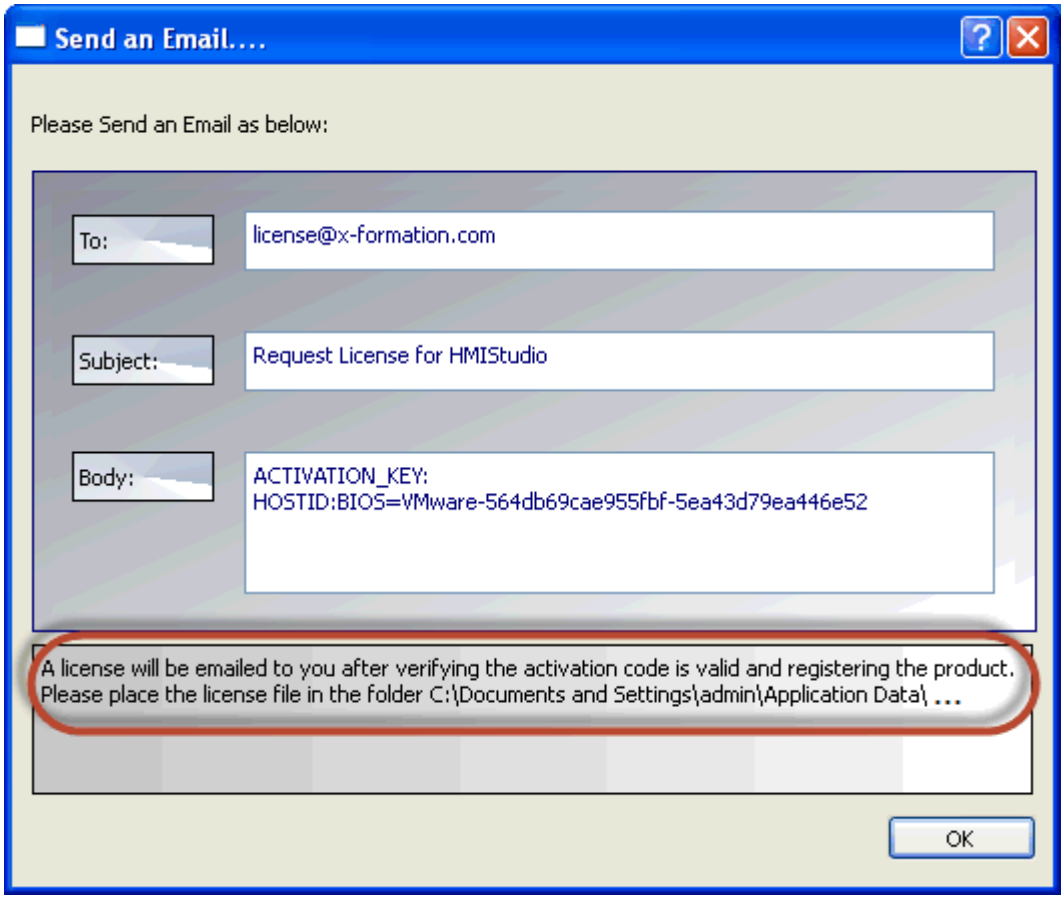

Send an Email....
Please Send an Email as below:
To:
license@x-formation.com
Subject:
Request License for HMIStudio
Body:
ACTIVATION_KEY:
HOSTID:BIOS=VMware-564db69cae955fbf-5ea43d79ea446e52
A license will be emailed to you after verifying the activation code is valid and registering the product.
Please place the license file in the folder C:\Documents and Settings\admin\Application Data\ ...
OK

Figure 8

To verify the status of an activation (in case of problems) is possible to introduce the Activation Key in the following URL: https://license.x-formation.com/licenses
From this website is possibile also to download the license file once activated and later save it in one of the following folders (based on OS):

- C:\Documents and Settings\[username]\Application Data\ Exor\
- C:\Users\[username]\AppData\Roaming\Exor\

NOTE *Virtual machines are not supported.*

1.3.4 Limitations

A table of Functions and Limits summarizes the limits for Runtime PC. In addition, the following features are not supported in Runtime PC:

Update package	creation of an update package for runtime & projects. However **Download to Target** via ethernet is supported.
Widgets	Video (analog video)
Manage Target	Board section is not supported
System Mode / User Mode	System Menu / TAP TAP & Rotating Menu not available
VNC/PDF readers	Standard PC software can be used in Runtime PC.
Backup/Restore	Not supported by Runtime PC. Standard PC software can be used for the purpose.
Tags	Max 4000 as reported in Tables and limits
Protocols	Runtime PC is compatibile with most Ethernet and RS-232 based serial protocols that do not require special hardware.

1.3.5 Fullscreen Mode

It is possible to choose if Runtime PC starts in fullscreen mode or in a window. To switch to fullscreen, right click with the mouse in the Runtime PC main window to go to the context menu and select Full Screen.

1.3.6 WORKSPACE

Using Runtime PC, project files are stored under workspace folder as per in panels.

Workspace folder s located under:

- %appdata%\Exor\[build number]\server\workspace

where [build number] is a folder named as build number like *01.90.00.608.*

1.3.7 FAQ: Why am I not able to download project to Runtime PC?

Runtime PC, by default, uses ports 80 and 21 for operations.
At startup, the system checks if the ports are already occupied by another service, if at least one is occupied it will prompt for port number change:

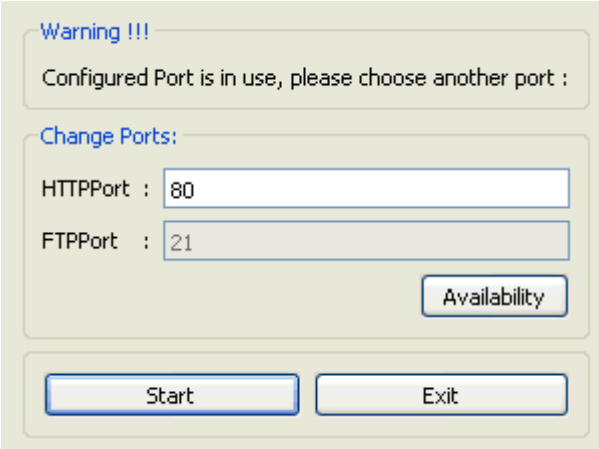


Figure 9

If you change listening port of your Runtime PC you must, by consequence, change downloading port used from *Download to target* functionality.
This is done through *Advanced...* menu:

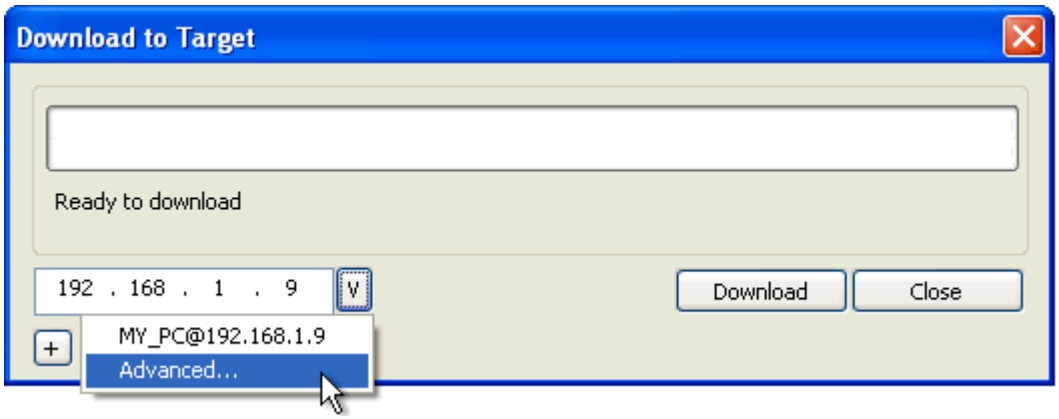


Figure 10

Here you must change port number accordingly to changes made on Runtime PC:

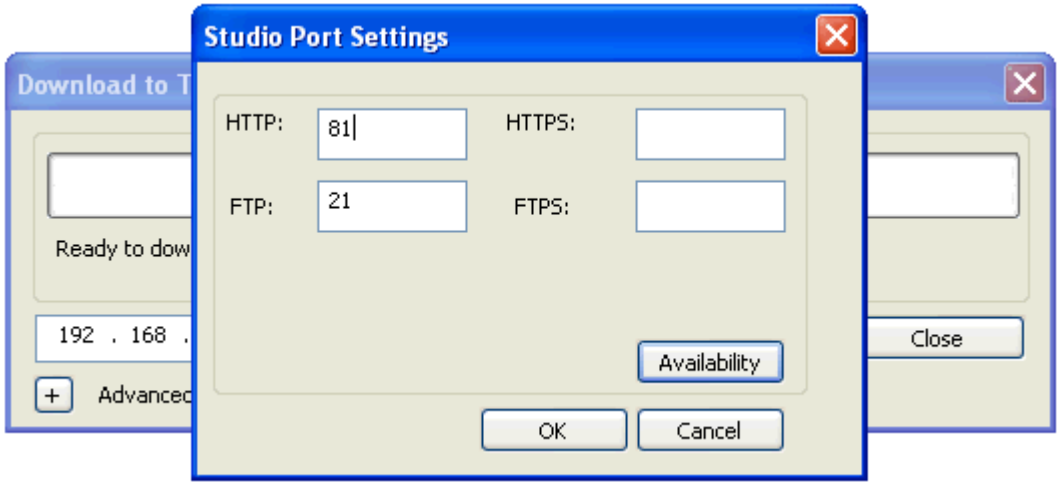


Figure 11

Once finished confirm with OK, then you are ready to download you project to the Runtime PC.

If you have lost information about changes made earlier on Runtime PC listening ports, when downloading project the system will return this error:

Impossible to establish communication with Runtime. Please check connection settings and verify the Runtime is properly running on target device.

To manually check/change current listening ports:

1) Shutdown Runtime PC
2) Locate folder: *%appdata%\Exor\[build number]\server\config*
   where [build number] is the the version of the product installed
3) Into this folder check two files:

- *httpd_config.xml*: here you will find current listening port for http service.
- *ftpdconfig.xml*: here you will find current listening port for ftp service.

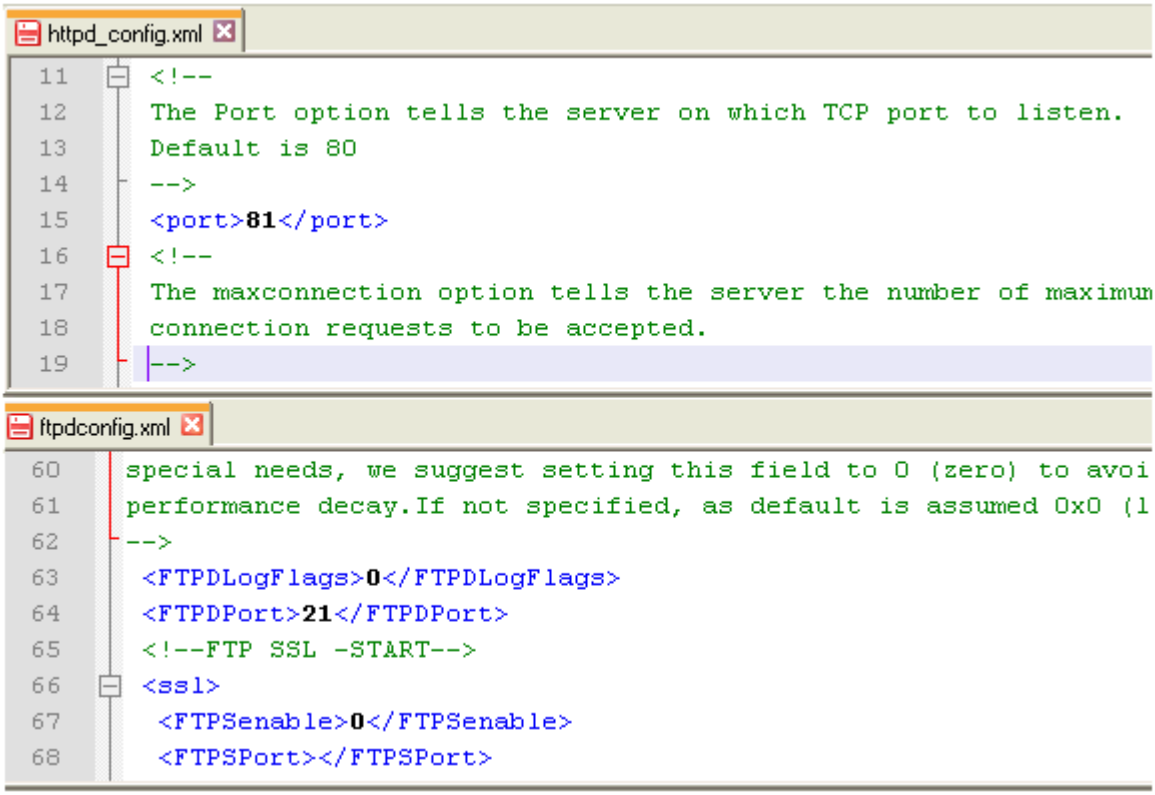


Figure 12

4) Now you can set into JMobile Studio the ports defined on these two files, as explained intoprevious chapter, to successfully download project to your Runtime PC.

If JMobile Studio is running on same machine where also Runtime PC is running, could happen that your personal firewall or antivirus blocks connection from JMobile Studio to Runtime PC.

Instead to select the machine IP address listed into discovery list, manually type-in the localhost IP address 127.0.0.1 in *Download to Target* dialog and select Download.

2 The HMI Runtime

The HMI Runtime is designed to support different platforms and different operating systems.
All the panels are running today on the base of the Windows CE operating system (Version 6 R3). The operating system and all its options are built around the minimum set of requirements of the HMI Runtime; there is no option to get direct access to the Operating system settings as all the needed components are managed via the runtime itself or via JMobile Studio.

Later in this document you will find more information on how to install the HMI Runtime and how to manage the update of other system components (firmware) on the units, but always with a dedicated interface which prevents a direct access to the operating system, often a source of complexity.

2.1 Runtime Modes

The HMI Runtime is composed of two logic units: the **server and** the **client**. The client unit is the part which is responsible for the visualization process: using the data collected at the server side to render it on the display as graphical information. The server unit is responsible for handling the HMI services such as running the communication protocols, performing data acquisition, driving trend buffer sampling activities, monitoring alarms, and so on. The server unit of the HMI Runtime can be in one of two operating modes:

- **Configuration Mode:** the server is idle; activity has not started; for example no project is loaded on the panel or system files are missing.
- **Operation Mode**: the server is active; it is operating according to the settings defined by the system files and by the application project.

The server operating mode is independent of the client side operating mode; you may have a visualization running but server activity stopped.

2.2 Basic Unit Settings

The settings of the device are available from the **Show system settings** menu, which is accessible from the Context Menu, if the panel has the runtime already installed, or by using the dedicated button on the unit when in loader mode (see for this the chapter "The Runtime Loader" below in this document).

Press and hold your finger on an empty area of the screen for few seconds, until the Context Menu appears as shown in the figure.

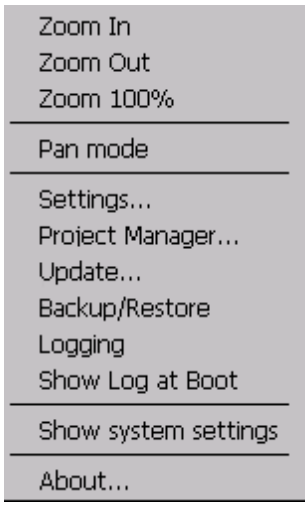


Figure 13

Select **Show system settings** to access the system settings tools.

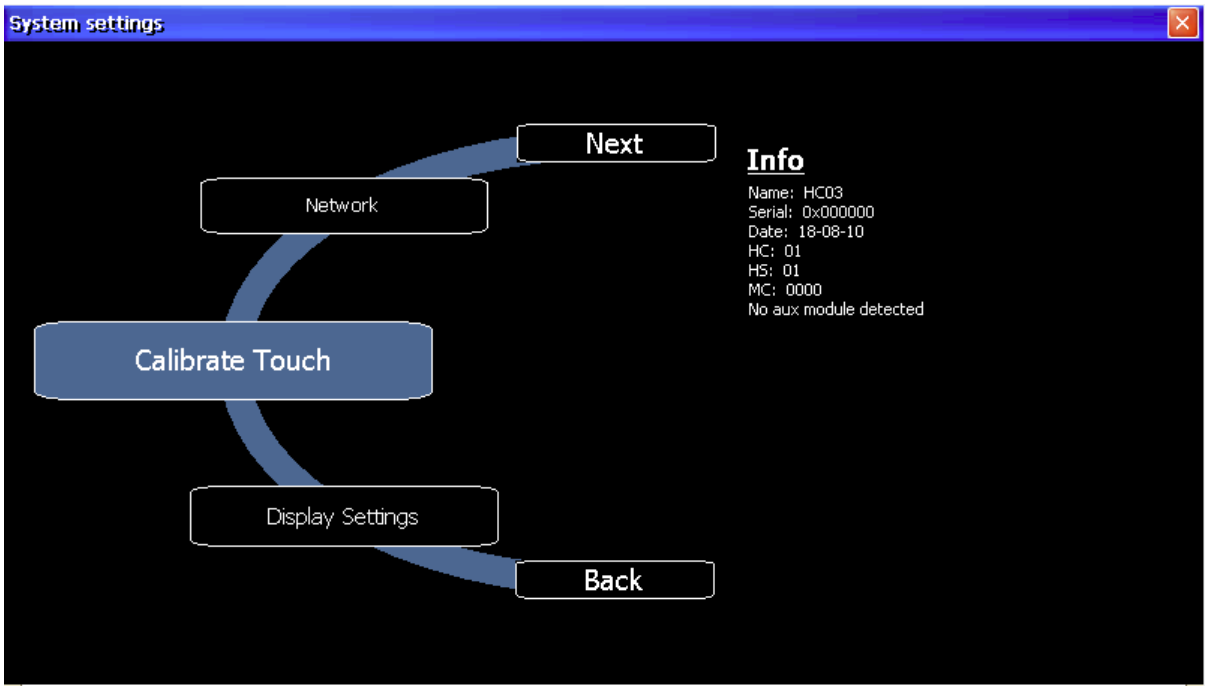


Figure 14

The System settings tool is a rotating menu through which you can scroll using the “Next" and “Back" buttons. It includes the following entries:

Calibrate Touch To calibrate the touch screen if needed

Display settings Backlight and Brightness control

Time Internal RTC settings

BSP Settings	Operating system version, Unit operating timers: power up and activated backlight timers, Buzzer control, Battery LED control
Network	IP address settings
Plug-in List	Provides a list of the plug-in modules installed and recognized by the system; this option may not be supported by all platforms and all versions.

2.3 Other Context Menu Options

The context menu has several other options:

Zoom In/Out/100%
Select view at runtime.

Pan Mode
Enables/disables pan mode; works only when you have previously activated a zoom in.

Settings
Following runtime settings are available:

Context Menu Delay (sec)	The context menu activation delay. Range is 1-60 seconds.
Show Busy Cursor	When enabled, shows an hourglass when the system is busy
Use keypads	When enabled, shows touch keypads when users touch/click on fields for data entry. When disabled, does not show any keypad on screen (useful when an external USB keyboard is connected to device).
Password	Change the password protecting operations such as: - Download Project/Runtime - Upload project - Board management (BSP Update) Please ref. to Remote access protection to HMI Panels for more information related to access protection.

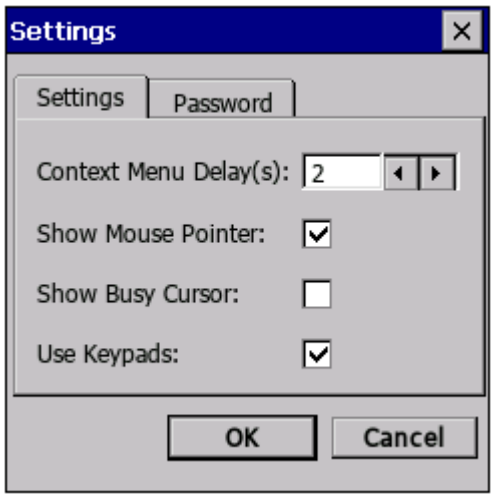


Figure 15

Project Manager
When activated, a dialog box will appear (see figure below) providing options to unload (de-activate) the current project, load (activate) another project present on the panel memory, or delete a project.
Please note that projects can be deleted only after they are unloaded. If you click on a project name other than the active one, the option "Load project" will first unload the running application and then automatically activate the new one.

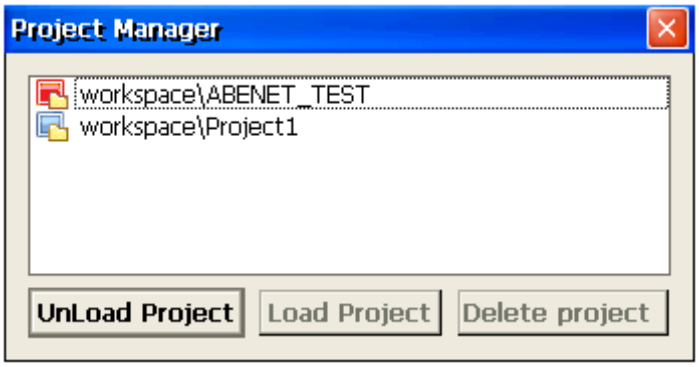


Figure 16

Update
When activated, the panel verifies first the presence of an external USB pen drive inserted in the panel USB port, and later for the presence on its root folder of the update package. See the “Updating Runtime from USB Pen Drive “chapter in this document for further information

Backup
Creates a backup copy of runtime and project.

Logging
Enables you to display a trace of the system operation log; it may be very useful in case there is a need to debug a problem of any nature. The following figure shows a case in which the system reports a communication error; the decoding of the reported information may not be immediate, but you can always use the option “Log to file” to save the dialog context to a file that can be later provided to Technical Support for investigation. The log file is called “logger.txt” and it saved to the folder “...\var\log” on the panel flash disk. The file can be retrieved from the panel using an FTP client.

NOTE *The “Log To File” Option is saved and retained after power cycles; when not needed any more, it must be manually deactivated.*

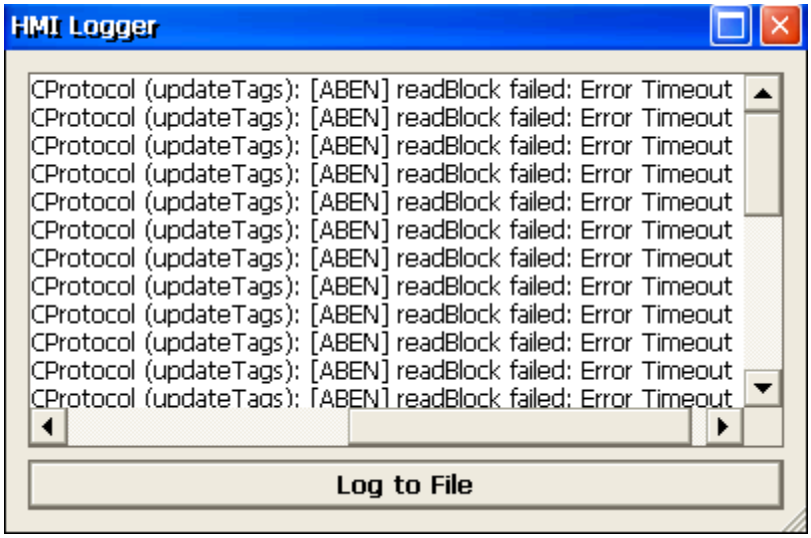


Figure 17

Show log at boot
Enables the logger at start up; if the “Log to file” option has been enabled, the files are saved, in this case, from the startup phase.

Developer tools
It is a collection of utility functions useful for debugging problems at runtime.

About
Shows information about the runtime version.

2.4 Built-in SNTP Service

The UniOP Panels Operating System features an integrated SNTP (**S**imple **N**etwork **T**ime **P**rotocol) that synchronizes the internal RTC panel whenever the predefined server is available.
The server addresses are hard-coded and cannot be changed by the user. The system searches for the following servers:

- *time.windows.com*
- *tock.usno.navy.mil*

SNTP servers are checked at power up, or once per week if the panel is not powered off.

NOTE *Starting from WCE v1.76 ARM / 2.79 MIPS is possible to customize up to two SNTP servers from System Mode (MainOS) accessible via Context Menu -> System Settings -> Time -> SNTP tab. This setting is not available in Configuration Mode (ConfigOS).*

3 My First Project

This section describes the steps to create a simple JMobile Studio project.

3.1 Creating a New Project

To create a new project, click on the **File > New Project** menu item.

The Project Wizard dialog will appear, asking for a project name and a path where the corresponding project folder will be stored.
JMobile Studio projects are stored in a folder that has the same name as the project. This folder contains all the files of the project. To move, copy or backup a project, you can simply move or copy the project folder and all its contents to the desired location.

NOTE *DO NOT rename the JMobile Studio Project folders manually. If you need to rename a project, use the **File > Save Project As** function. Depending on the size of the project, this could take some time.*

Click **Next** to go to the panel selection dialog.

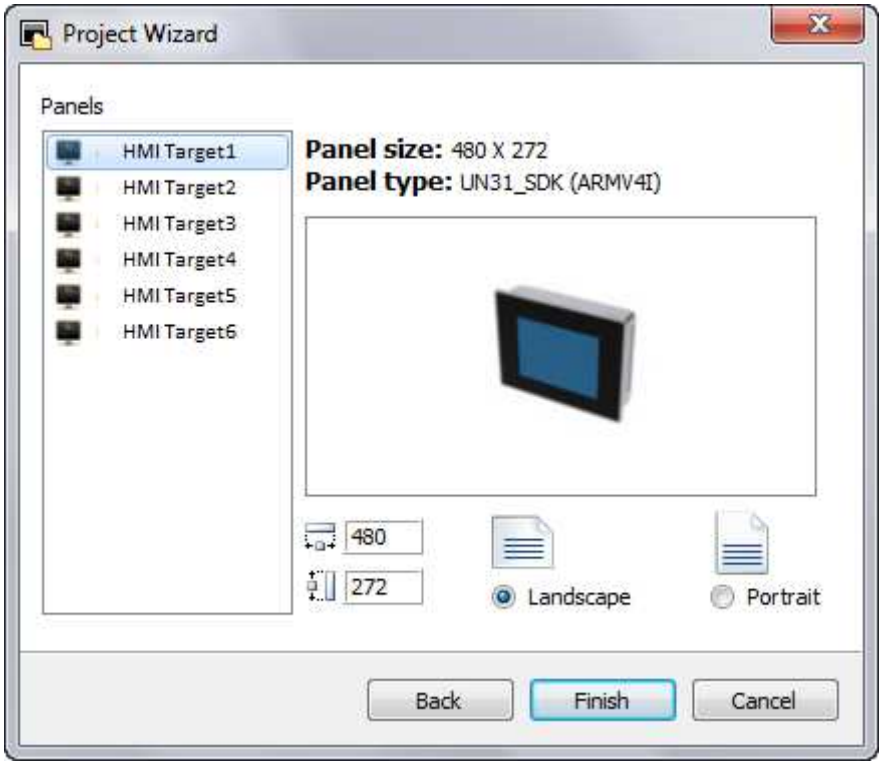


Figure 18

The panel selection is shown in the figure above. Here, you can scroll through a list of available HMI models to select the model you are working with.

For each model, two radio buttons are available to select the orientation: landscape (default) or portrait. In portrait mode the device is rotated 90° clockwise.

Some software features are not rotated when portrait mode has been selected. A list of these features is shown in the table below.

WCE dialogs	all dialogs related to “System Settings”
System Dialogs	ex. System Mode
ContextMenu and related dialogs	Project Manager, About, Settings, Logging, Backup
Video	Analog Video Input, IPCamera, MediaPlayer
JavaScript	Alert and Print function
Dialog pages	“Title” of dialog pages
Scheduler	Dialogs for data entry do not support portrait
Macro	ShowMessage, LunchApplication, LunchBrowser
External applications	PDF Reader, VNC

Click **Finish** to complete the Wizard.

Once the HMI model is chosen, you can convert the project to any other model, using the project properties portion of the screen, as shown below. This will not resize all widgets in the project to the correct size to fit a smaller or larger screen; it will simply change the model type and give a warning if some objects will be lost during the conversion.

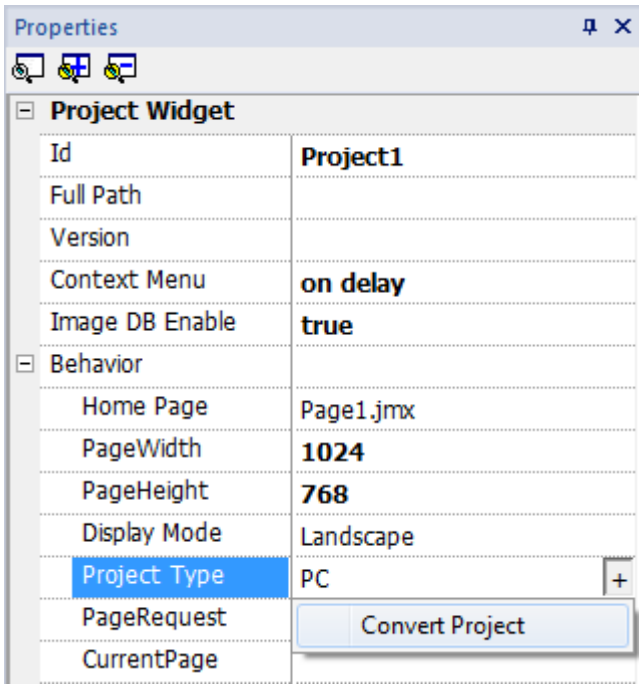


Figure 19

3.2 The Workspace

The JMobile Studio workspace is divided into following main areas:

Project View	Presents the elements of theprojec in the form of a hierarchical Project Tree.
Object View	Lists the Widgets with the corresponding ID's used in the page.
Working Area	Main working space where editors create the HMI pages. The current page or pages opened in the Editor View are indicated by a tab at the top of the center area. You can quickly switch between the different pages in the Editor View by clicking on the desired tab.
Properties	Properties for the selected object / widgets
Widget Gallery	Large library of symbols and graphic objects.

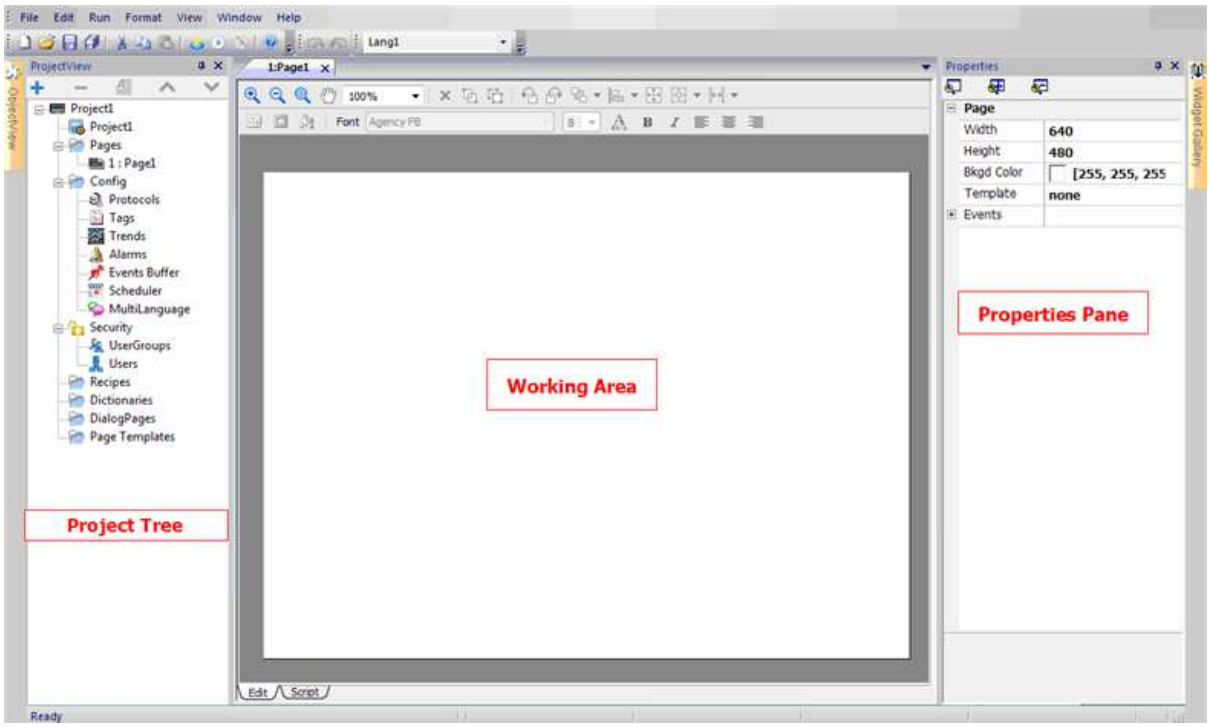


Figure 20

The workspace layout can be freely changed at any time; any change is saved and maintained among Studio activations. In case you need to reset the workspace to the original default layout, use the command called **Reset and Restart** under the File menu.

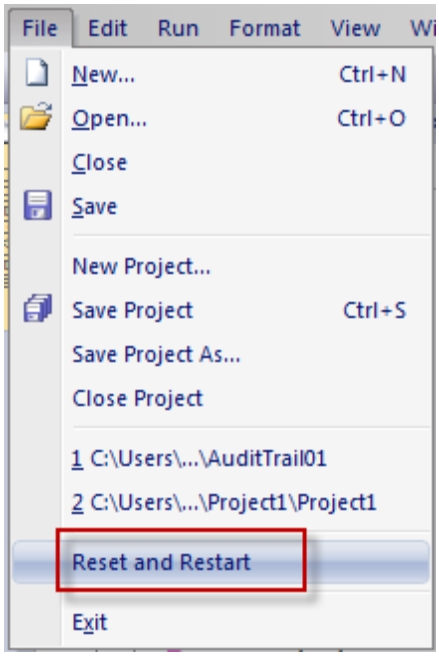


Figure 21

3.3 Communication Protocols

Device Communication drivers are configured in the **Protocol Editor**, which is accessible from the project tree (as shown in the figure below). Double click on the Protocols icon in the Project Tree view to open the Protocol Editor.

To add a driver, click on the “+" Icon and select the driver from the list in the controller field. Once a communication driver has been selected, configure the driver by clicking on the browse button in the column Configuration. A configuration dialog will be displayed, allowing you to set the parameters of the driver (as shown in the figure below).

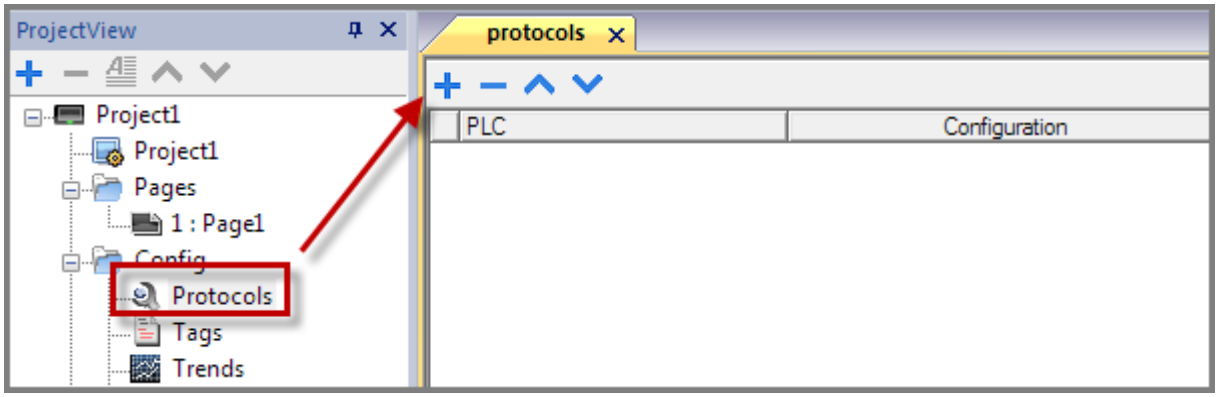


Figure 22

As an example, to create a project for Modbus TCP, you would select the Modbus TCP driver and then configure the communication parameters by selecting the browse button in the Configuration column.

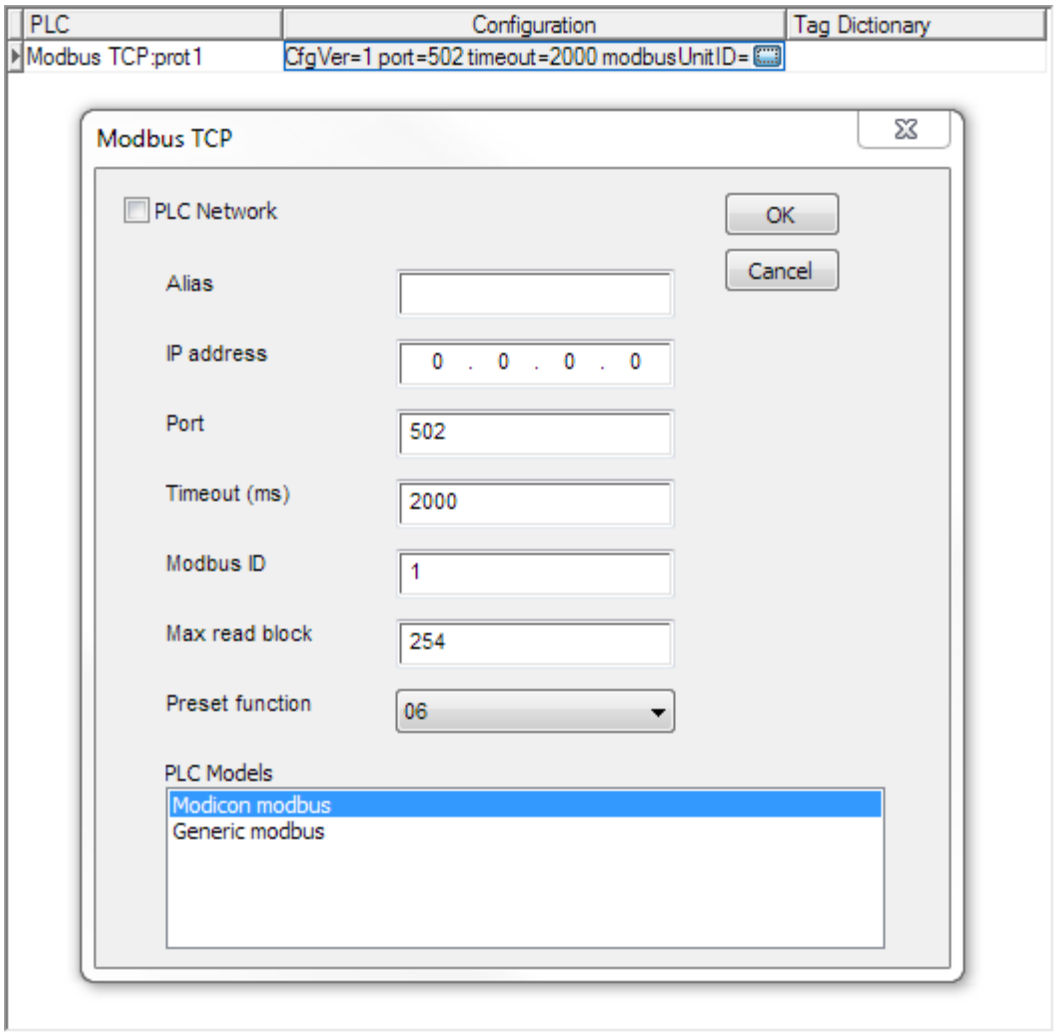


Figure 23

JMobile Studio configurations that include more than one communication protocol. By repeating the steps previously outlined, you can add up to four protocols in the Protocol Editor.

NOTE *while it is possible to run different Ethernet protocols over the same physical Ethernet port, you cannot run different serial protocols over a single serial port. Some serial protocols support access to multiple PLCs, but this is an option that has to be configured within the protocol and still counts as one protocol.*

The Other parameters in protocol editor are:

- **Tag Dictionary**: tags imported for a particular protocol. Ref. to chapter **Dictionaries** for more details
- **Enable Offline Algorithm / Offline Retry Timeout:** ref. to chapter **Offline Node Management** for more details
- **Version:** version of the protocol available in JMobile Studio for selected target. Version of the protocol is not read from hmi directly but from studio internal DLLs.

3.4 Tags

JMobile Studio uses Tag names to access all device data. All fields and reference locations in the device need to be assigned a Tag name to be used in the HMI. To assign Tags, double click on the Tags icon in the Project View and the Tag Editor will be displayed (as shown in the figure below).

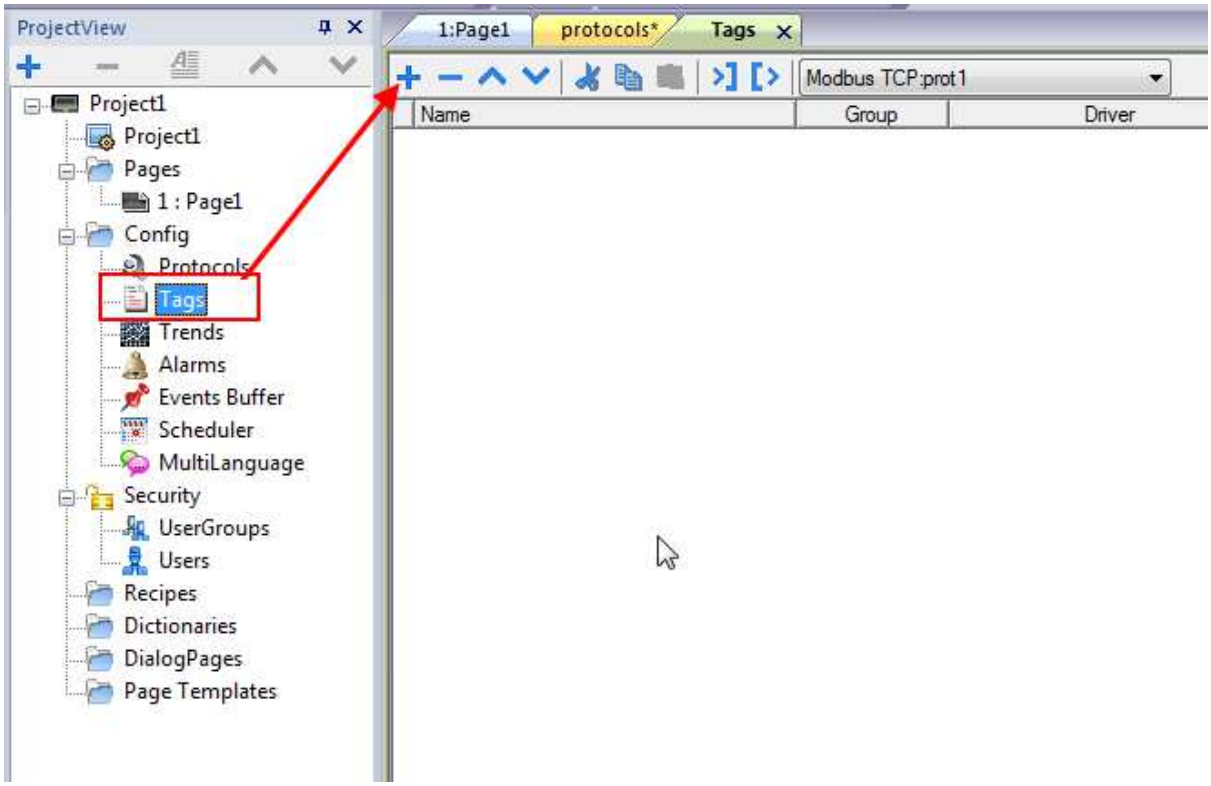


Figure 24

To add a new Tag, click on the “+" icon, and select the Address from the Communication protocol address dialog. When Tags are initially added, these Tags are named Tag1, Tag2, etc., by default. The user can rename the Tag with the appropriate name by clicking once on the Tag name.

The Tag Editor provides a **Tag Import** feature, which is available based on the protocol selected. Not all protocols support Tag Import. If the protocol does support this feature (see specific Protocol documentation), first select the Protocol from the filter button and then click on the Import button (as shown in the figure below).

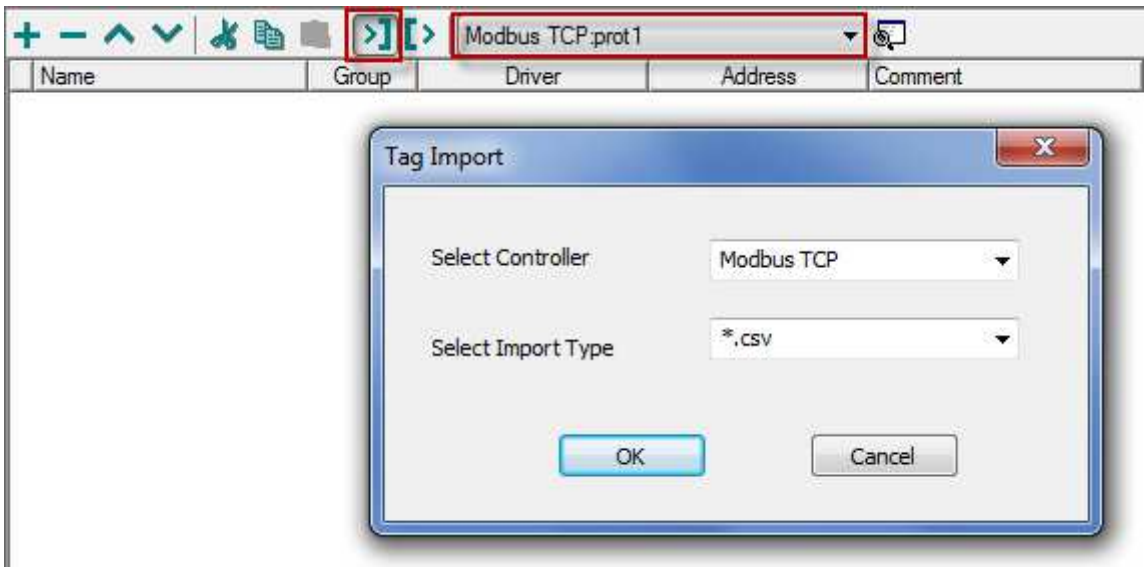


Figure 25

You will see the dialog that corresponds to the protocol selected, which prompts you to browse for the symbol file. The symbol file is exported from the controller programming software.

3.4.1 Tag Editor

The tool in JMobile Studio used to create and manage tags is called the Tag Editor.

For each tag, the Tag Editor allows you to specify several properties:

Name
This is the unique name at the project level of the tag. This is the primary key used to identify the information in the internal runtime tag database. Note that you cannot use the same tag name even if you are referring to different communication protocols.

Group
After the tags have been defined in the Tag Editor, they are used in the project by attaching them to the widgets' properties (see chapter "Attach To" for a complete explanation).
For each screen the system is able to identify which tags are used in the specific page and identifies them as part of the "page group". This allows easy handling at run time of the requests made by the communication protocol to the connected controller(s): only the tags included in the displayed page are queued for polling from the controller memory.
This mechanism is fully automatic and there is no intervention required by the user.
The tag editor allows you to define groups of tags not belonging to a specific page but, for instance, grouped according to their logical meaning.
We can call these groups "Users' groups". Users' groups have no meaning for the local visualization, but they are very useful when external software communicates with the runtime requesting sets of data that must be independent from the currently displayed screen.
The Runtime web server publishes a set of communication interfaces that can be used from a 3rd party application to interface with the local tag database and read the tags according to their grouping.
The group column allows you to define the users' groups and assign tags to them.

Driver
Specifies the communication protocol for which the tag is defined.

Address
This shows the PLC controller memory address. To edit it, click on the right side of the column to get the dialog box where you can enter the address information.

Encoding
Encoding type for string data type (UTF-8, Latin1, UTF-2 and UTF-16)

Comment
Allows you to add a description of the tag.

Rate (ms)
Define the refresh time for tag. Default is 500ms that means tag is updated every 500ms.

R/W
This option determines if the tag must be managed as Read only (R), Write only (W) or Read/Write (R/W). When a Tag is Write Only (W), the system never reads the tag value (& status) but can only write it. When communication is not active, content of Write Only tags may not be available in widgets.

Active
As explained above, tags are grouped per page and, if needed, in users' groups. By default, tags are not active (are not read from the controller); this means they are automatically activated by the runtime when the visualization requires them (for example when used on current page). You can force the system to

continuously read a certain tag even if not present in the current page by setting its Active property to true. We recommend that you normally leave this parameter to false to avoid unexpected results in terms of communication performance.

Simulator
JMobile Studio provides online and off-line simulation. The behavior of each tag during offline simulation mode can be specified by choosing between several profiles.

Scaling
Tag values are normally transferred "as they are" from the protocol to the real time tag database. You can specifically apply scaling to the tag values before they are stored in the database. The available scaling options are **By formula** and **By range**. Scaling can be specified in terms of linear relationship as a formula or as range conversion.

The tag name must be always unique at the project level; often it may happen that the same tags, from the same symbol file have to be used for two different controllers. Since having tags with the same name is not supported, you can use the “Alias” feature to automatically add a prefix to the imported tag to make them unique at the project level.
When importing tags for a Protocol, the tag names may be prefixed by the name given in the “Alias” item of the protocol configuration dialog box. Please note that not all protocols support the “Alias” feature. See protocol documentation for specific information.

PLC Tag Name
This is the original name of the tag when imported from PLC. This field is managed automatically by Tag Importer and is available as R/W in advanced view just to allow users to change it if any problem during tag import operation. For tags not imported from external files usually this field is empty and can be ignored. PLC Tag Name are used so as link between tags used by HMI application (Tag Name) and tags exported from PLC.

3.4.2 Data Types

When creating a tag, JMobile Studio shows a dialog box in which you need to specify the tag properties. The tag Memory Types are specific to the selected Protocol.

The tag Data Type must be selected from the list of available JMobile Studio Data Types, according to the internal representation you need for the selected controller address. JMobile Studio Data Types are summarized in the following table.

Data Type	Description
string	Character strings. The characters are coded in UTF-8 format.
boolean	Boolean is one bit data (0..1)
float	Float corresponds to the IEEE single-precision 32-bit floating point type (1.17e-38 ... 3.40e38)
double	Double corresponds to IEEE double-precision 64-bit floating point type (2.2e-308 ... 1.79e308)
binary	Binary represents arbitrary binary data
int	Int is signed 32 bit data (-2.1e9 ... 2.1e9)

short	Short is signed 16 bits data (-32768..32767)
byte	Byte is signed 8 bits data (-128..127)
unsignedInt	UnsignedInt is unsigned 32 bit data (0 ... 4.2e9)
unsignedShort	UnsignedShort is unsigned 16 bit data (0..65535)
unsignedByte	UnsignedByte is unsigned 8 bit data (0..255)
time	Time data
boolean [ ]	Array of Boolean
byte [ ]	Array of byte
short [ ]	Array of short
int [ ]	Array of int
unsignedbyte [ ]	Array of unsignedbyte
unsignedshort [ ]	Array of unsignedshort
unsignedint [ ]	Array of unsignedint
float [ ]	Array of float
double [ ]	Array of double
time [ ]	Array of time

3.4.3 Dictionaries

A dictionary is a list of tags imported in the tag editor for a specific protocol. Usually these files are generated by 3d party tools and are in .csv, .xml or other formats. Refer to **Tag Import** section of each protocol for details related to supported formats.

Dictionaries folder in ProjectView list all files imported in the tag editor for each protocol. Selecting a particular protocol, it is possible to delete or look at the imported dictionary files for the related protocol.

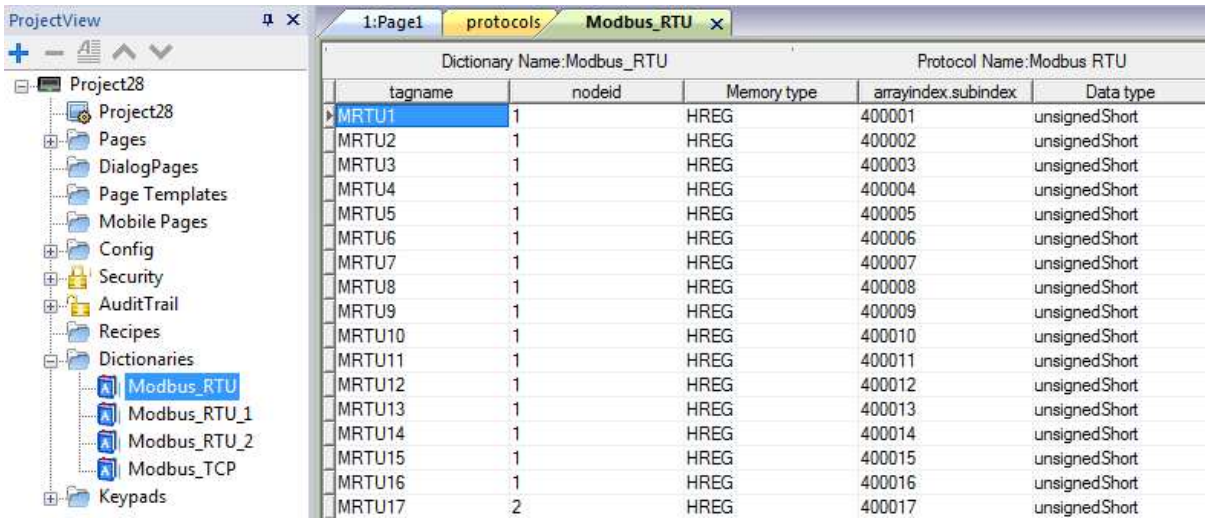

ProjectView
Project28
Project28
Pages
DialogPages
Page Templates
Mobile Pages
Config
Security
AuditTrail
Recipes
Dictionaries
Modbus_RTU
Modbus_RTU_1
Modbus_RTU_2
Modbus_TCP
Keypads
1:Page1
protocols
Modbus_RTU
Dictionary Name:Modbus_RTU
Protocol Name:Modbus RTU
tagname | nodeid | Memory type | arrayindex.subindex | Data type
MRTU1 | 1 | HREG | 400001 | unsignedShort
MRTU2 | 1 | HREG | 400002 | unsignedShort
MRTU3 | 1 | HREG | 400003 | unsignedShort
MRTU4 | 1 | HREG | 400004 | unsignedShort
MRTU5 | 1 | HREG | 400005 | unsignedShort
MRTU6 | 1 | HREG | 400006 | unsignedShort
MRTU7 | 1 | HREG | 400007 | unsignedShort
MRTU8 | 1 | HREG | 400008 | unsignedShort
MRTU9 | 1 | HREG | 400009 | unsignedShort
MRTU10 | 1 | HREG | 400010 | unsignedShort
MRTU11 | 1 | HREG | 400011 | unsignedShort
MRTU12 | 1 | HREG | 400012 | unsignedShort
MRTU13 | 1 | HREG | 400013 | unsignedShort
MRTU14 | 1 | HREG | 400014 | unsignedShort
MRTU15 | 1 | HREG | 400015 | unsignedShort
MRTU16 | 1 | HREG | 400016 | unsignedShort
MRTU17 | 2 | HREG | 400017 | unsignedShort

Figure 26

To import a new Dictionary proceed as follow in Tag Editor:

1. From top toolbar, select interested protocol
2. Click on **>]** button to call importer
3. Verify controller type and select format of file to import (.csv, .txt, .sym – protocol dependent)
4. Click **OK**
5. Select file to import

As result, a new dictionary file is added to the **Dictionaries** folder and a list of tags imported are available in the tag editor and shown at the bottom of the tag editor page. Tags shown in dictionary can be imported into the project using following:

- Import Tags (to add new tags to the project)
- Update Tags (to update tags already imported previously)

NOTE When importing tags, the "." period is replaced with a "/" forward slash character. This is normal and the protocol will use the correct syntax when communicating to the PLC. The "." is a reserved character and cannot be used in a tag name.

NOTE The “&” ampersand character cannot be used in a tag name, as it can cause communication issues.

3.5 Designing a Page

When a project is created, a page is automatically added to the project and shown in the Page Editor. To add objects to a page, simply drag and drop the objects from the Widget Gallery to the page.

To add a new page, right click on the Page node from the project tree and select “Insert new page”. A dialog box will appear asking for the name of the new page.

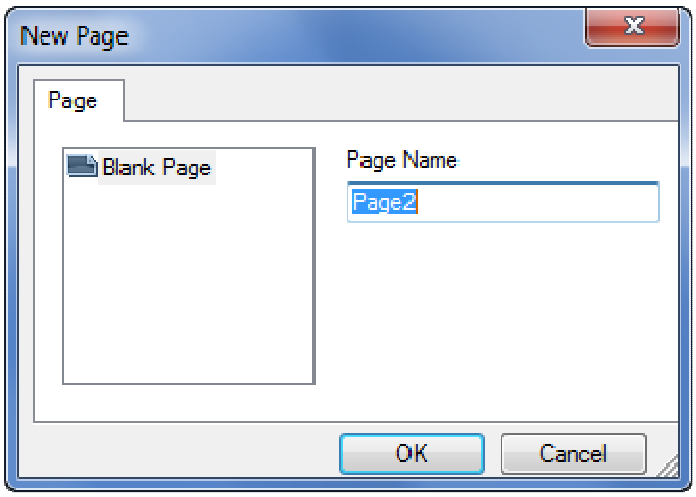


Figure 27

3.5.1 Importing a Page

A page can also be imported from another project. By right clicking on the page folder in the Project View, you will see an option named “Import Page”. Please refer to the Figure below.

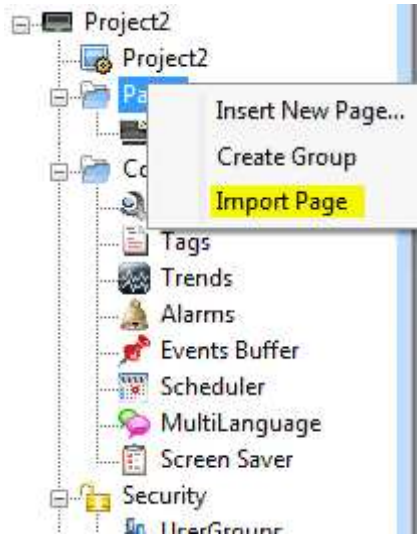


Figure 28

After selecting a page to be imported from the desired project, when you click OK, you get a warning message in the editor as shown in the figure below.

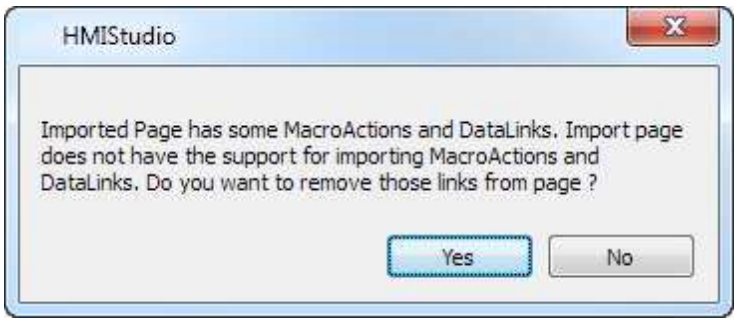

HMIStudio
Imported Page has some MacroActions and DataLinks. Import page does not have the support for importing MacroActions and DataLinks. Do you want to remove those links from page ?
Yes
No

Figure 29

Page Import can support only import of the page and the widgets in it, but not the Macro actions and data links attached to the widget. By selecting “Yes” all the data links and the Macro actions attached to the widgets will be removed. Only the Widgets will remain. By selecting “No” the macro actions and the data links will remain attached to the widgets, but may not function properly during runtime, unless the tags associated with the Macros and data links are present or created in the new project.

NOTE *The page import can be done between projects made in the same version of the software. If the versions are different, a warning message will pop up to save the project in the new version, and then try again to import the page.*

3.5.2 Dialog Pages

Dialog pages are windows opened at runtime on top of the current page when requested by the application. Dialogs are used to inform user about something happening (ex. alarms/notifications/status errors) or to allow user to answer a question.

Dialog type can be defined in the property window of each dialog and can be:

- **Modal:** user cannot return to main project window/page until dialog is closed.
- **Non-Modal:** user can continue to use main project window (or other dialogs non- modal) while a dialog is shown on top of it.

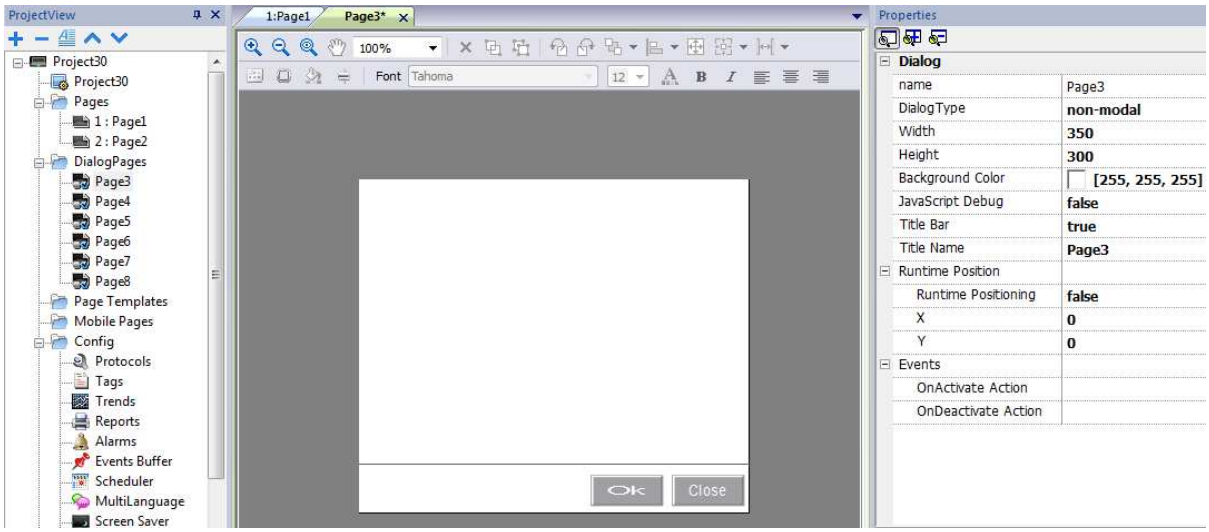


Figure 30

Max number of dialogs allowed is reported in **Table of functions and limits.** When max number of open dialog has been reached, the runtime will close automatically the oldest dialog open to open the new one.

A dialog can have a **Title Bar** on top of it. When Title Bar is enabled (**Title Bar = true),** a **Title Name** may be shown.

Runtime Position can be used to specify a fixed position for the dialog window.

3.6 The Widget Gallery

The Gallery is adjacent to the Property View panel and can be opened by clicking on the Widget Gallery tab (as shown in the figure below).

Figure 31

Select the desired object from the Widget Gallery, then drag and drop it on the page.
To change the appearance of the object, select the desired property from the property pane and change the property settings.

All the HMI objects required to build an application are available in the Widget Gallery. The Widget Gallery is accessible as a slide in pane from the right side of the workspace (as explained in the previous chapter).
The gallery is divided into several categories, each with collections of different types of objects.
Click on a category to display its sub-categories.
For each sub-category, the gallery offers the option of applying different styles to the objects within that category (when possible).

The figure below shows the Widget style button for round gauges.

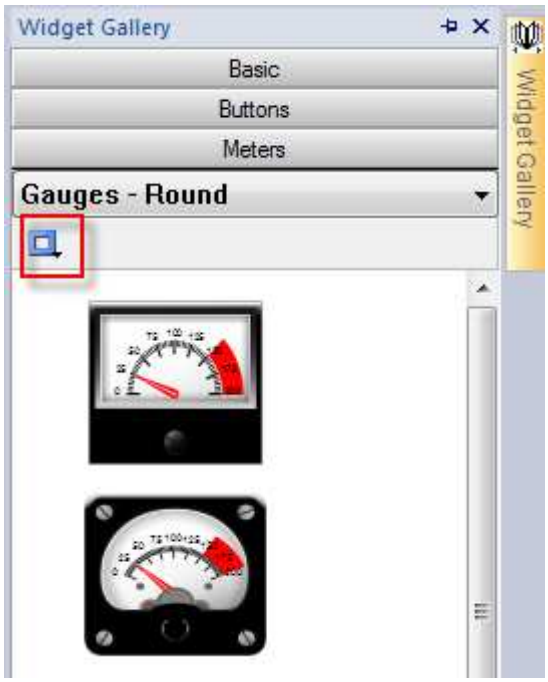


Figure 32

Clicking on the style button will display the available styles for the current object.
Select one of the available styles to apply it to the gallery objects.
This is done using the Page Toolbar shown in the figure below.

NOTE *Style change may not be available for all widgets.*

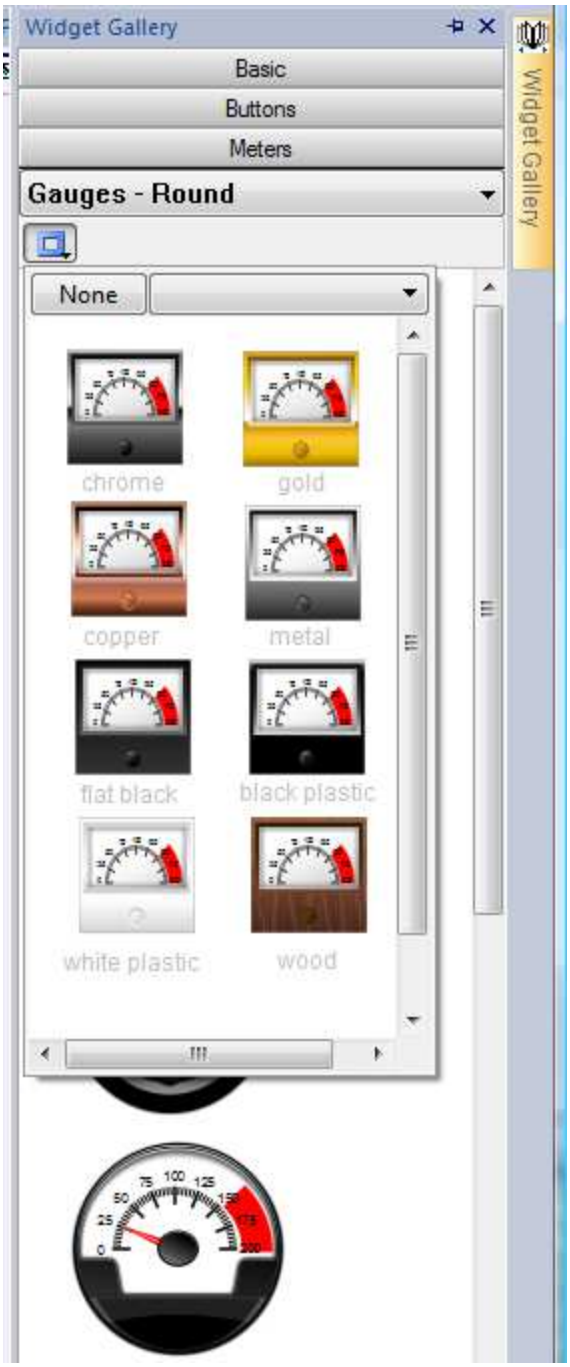


Figure 33

Once on the page, the object can still be subject to additional style changes.
This is done using the Page Toolbar shown in the figure below.
Depending on the object selected, you can have options for the style, frame, fill color…along with the font type and size and other standard object properties

NOTE *Some widgets are composed of many sub widgets. For example a button is a complex widget composed by two Image widgets, a button widget and label. This is clearly visible in the* ***ObjectView*** *when the widget is selected. To select a sub widget like the label in a button, use ObjectView or* ***Shift + leftClick*** *of mouse. In this way sub widget can be changed without ungroup all widget.*

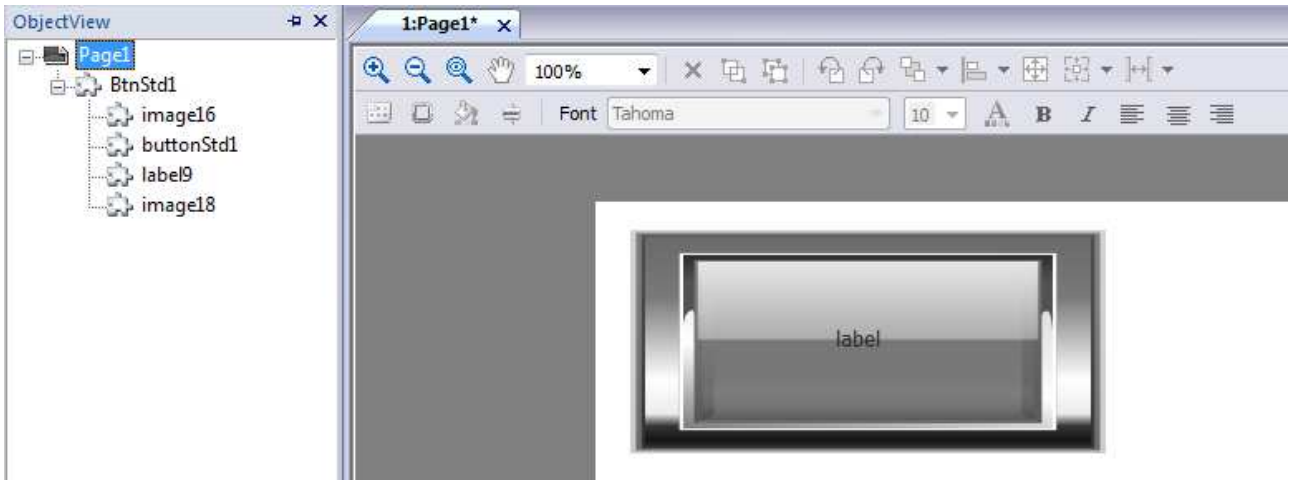


Figure 34

3.7 "Attach To" and Dynamic Properties

JMobile Studio allows for simple binding between Tags and Widget Properties. Many different Widget Properties can be attached to a Tag, which allows you to control the device and animate objects based on live data.

To attach a Tag to a property, click on the property in Property view. A **[+]** button will be displayed on the right side of the property. Click on this button and select the item **Attach To…** from the menu (as shown in the figure below).

For example, when working with a gauge object, the most common action taken by the programmer is to attach a Tag to the needle, so that the value of the Tag referenced in the controller memory is represented by the needle movement.

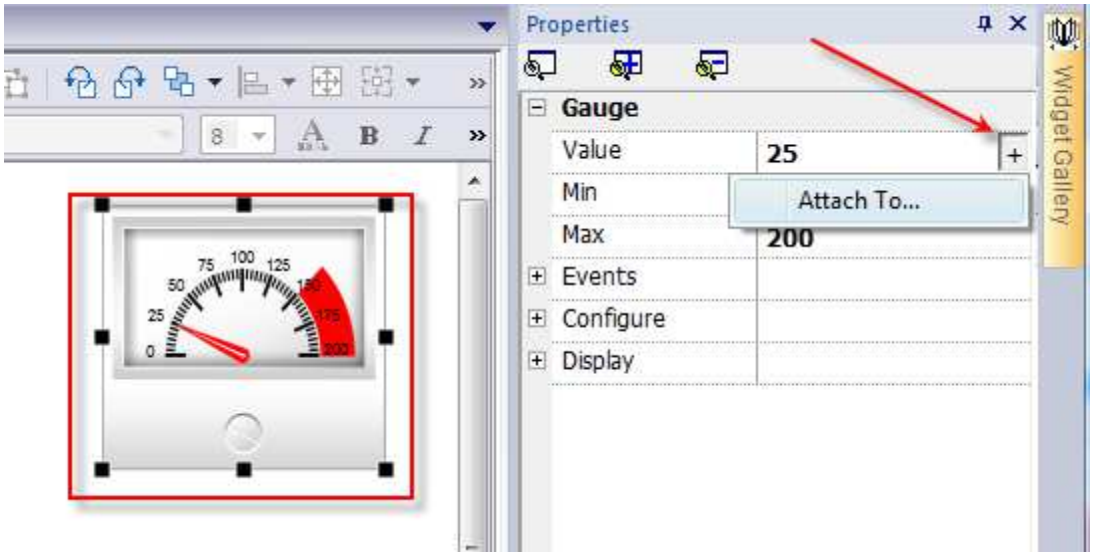


Figure 35

To attach the Tag to the needle, single click on the object to display its properties in the Property view. Locate the **Value** property and click on the [**+]** button on the right part of the field as shown in above figure. Select the **Attach To…** menu item and a dialog will be displayed as shown in the figure below.

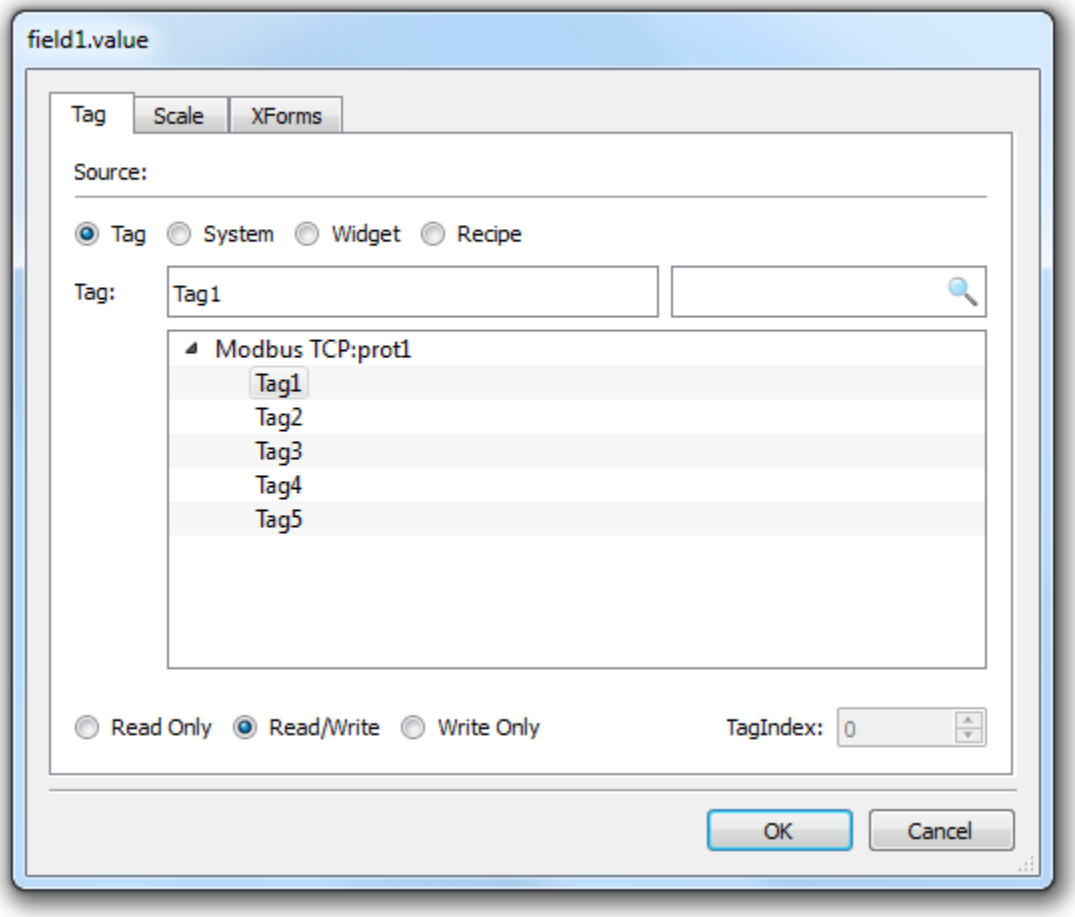


Figure 36

When attaching a Tag, you can attach four types of data sources:

- **Tag**: tag defined in the Tag editor
- **System**: predefined system tags (example date and time)
- **Widget**: connect to a widget property (example: value of a slider widget)
- **Recipe**: recipe data from Recipe Manager

Select the Tag from the Tag list and Click OK to confirm.

Tags can be attached to many different properties of the object. You can attach a Tag to a property by selecting the property in the Property view and clicking on the Attach To or you can right-click on the object and select the Attach To… menu item.

NOTE *A chapter in this document describes in detail the "Attach to" concept.*

4 The HMI Simulator

The HMISimulator provides the ability to test the project functionality before downloading it to the panel. This feature is useful to test the project when no PLC or HMI hardware is available and to speed up development and debugging of projects.

The HMI Simulator support online simulation in communication with real devices (PLC based on Ethernet or RS-232 based protocols) and offline simulation (where using **Tag Editor** -> **Simulator** field allows the configuration behavior of each tag in simulation mode).

4.1 Launching the Simulator

Start the Simulator with the menu item **Run > Start Simulator**.

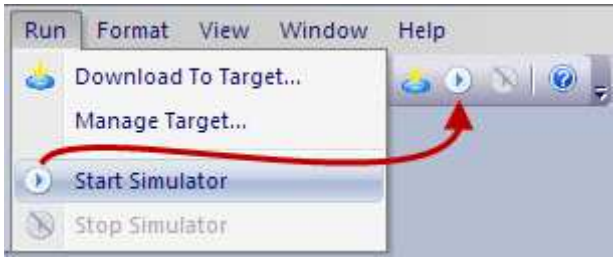


Figure 37

At this point, the Simulator is running in the computer, similar to the way the server runs on a panel.

4.2 Stopping the Simulator

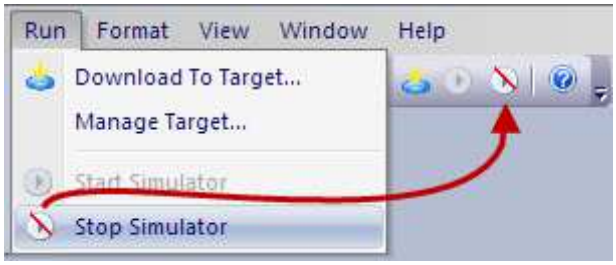


Figure 38

To stop the Simulator, select the **Run >Stop Simulator** menu item. You can also exit the Simulator using the close button of the Simulator or by using the **Exit** option from the Context Menu.

4.3 Simulator Settings

The Simulator can be used with real protocols & PLCs (Ethernet or RS-232 based protocols) or with simulated protocols.

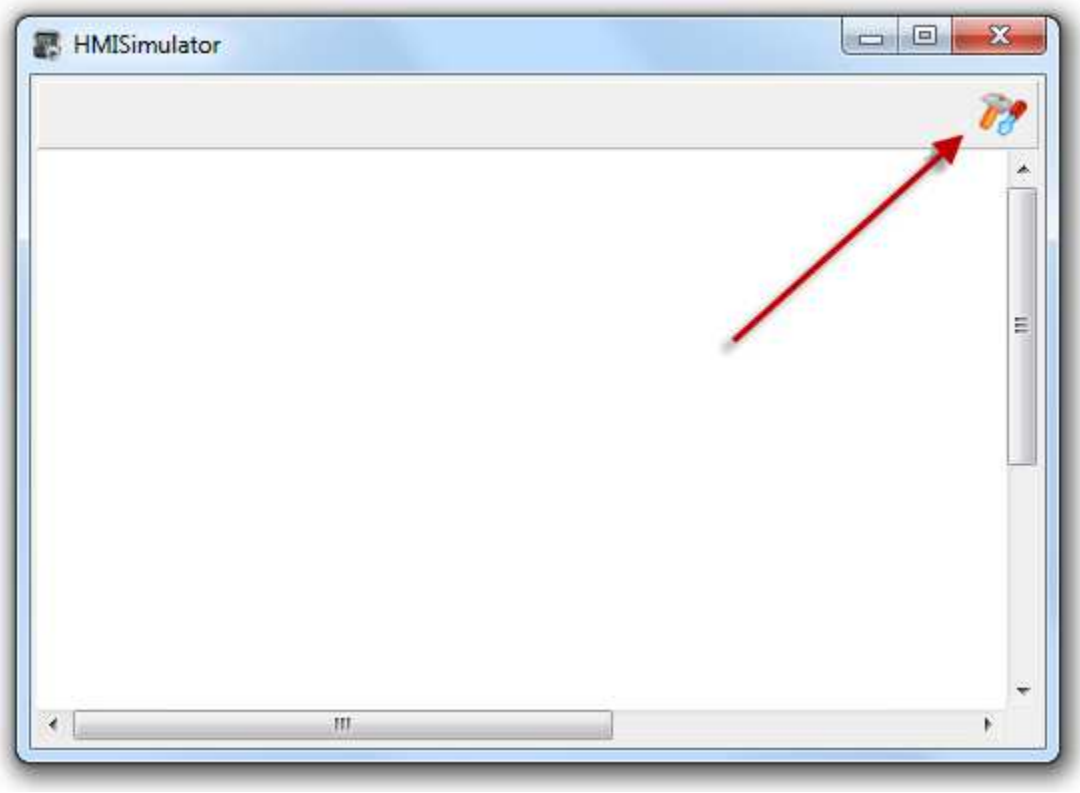


Figure 39

When we invoke the **Simulator Settings** button, a dialog showing the protocols used in the project will pop-up. Users can select to use actual or simulated protocols by using the **Use Simulation** checkbox.

By default the Simulator uses simulated protocols defined in the *Tag Editor* **Simulator** column for each Tag (see Figure 40). Unchecking flag **Mode,** the Simulator will communicate using real protocols with devices.

NOTE *Some protocols, for example the Variables protocol, does not support communication with devices (Win32) and, for these protocols, this option remains disabled. Usually all protocols based on ETH or RS-232 can be simulated in win32 platform or in general all protocols that do not require special hardware.*

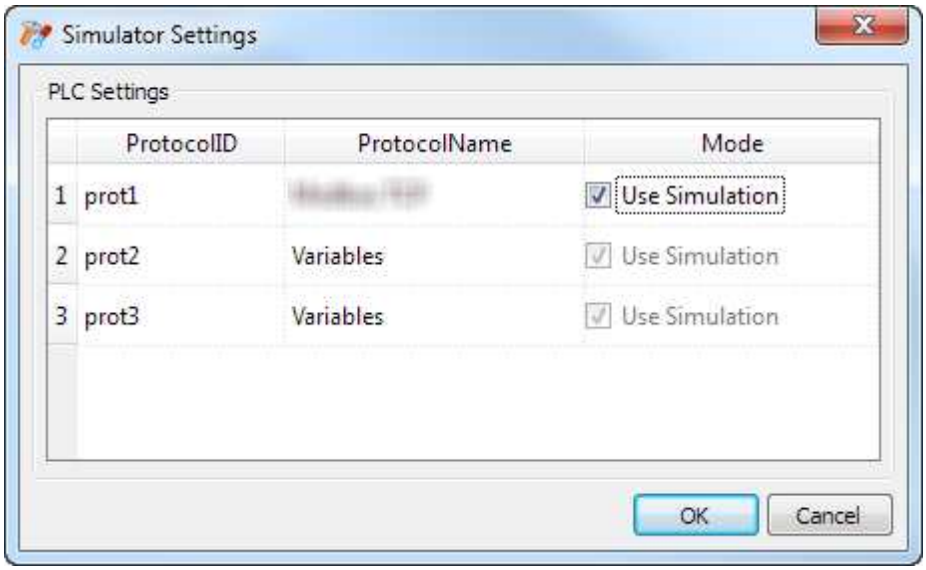


Figure 40

When defining Tag values, the *Tag Editor* also includes a field *Simulator* to select a method for simulating the data. Tag values can be simulated in the following ways:

Variables	The data is stored in a variable in the simulator. This variable holds the value of the Tag so the client can read and write to the Tag value.
SawTooth	A count value is incremented from “Offset” to “Amplitude + Offset” value with a “Period” of 60..3600 seconds. When the counter reaches “Amplitude + Offset”, the value is reset to “Offset” and the counter restarts.
Sine Wave	A sine wave value is generated and written to the Tag value. The **Min**, **Max** and **Period** values of the Sine wave can be defined for each Tag.
Triangle Wave	A triangle wave value is generated and written to the Tag value. The **Min**, **Max** and **Period** values of the wave can be defined for each Tag.
Square Wave	A square wave value is generated and written to the Tag value. The **Min**, **Max** and **Period** values of the Sine wave can be defined for each Tag.

5 Transferring the Project to Target

The JMobile Studio project can be transferred to the runtime in two ways:

1. Using the **Download to Target** item in the Run Menu.
2. using an **Update Package** via USB

5.1 Download to Target

Run -> Download to Target can be used to transfer project and runtime via Ethernet from the JMobile Studio to the runtime.

NOTE *The panel must have a valid IP address assigned. Please see the chapter "Unit Basic Settings" for further information on how to assign an IP address to the panel.*

Once the panel has a valid IP address assigned, it will become discoverable on the local network. Click on the **discovery** button and select the HMI from the list of IP addresses.

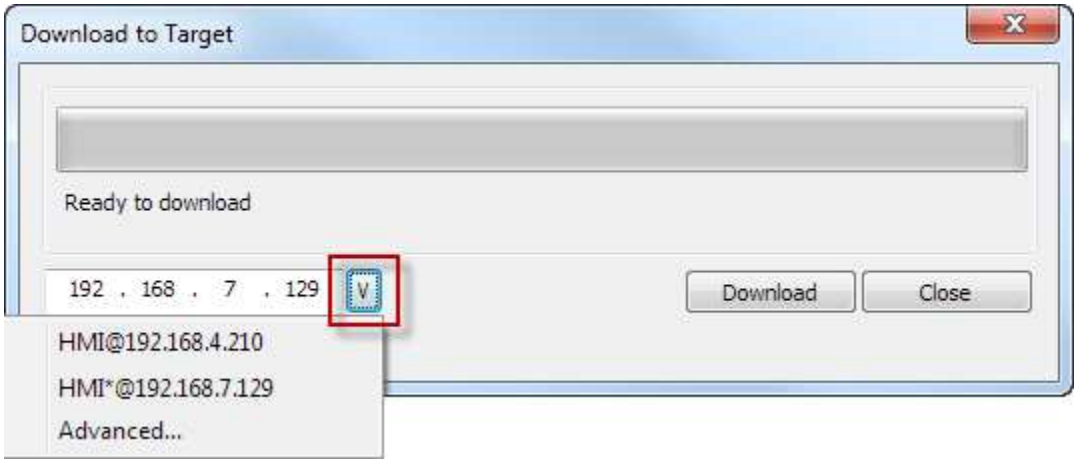


Figure 41

Click on the **Download** button to start the process. The system will switch the Target to Configuration mode and transfer the files. When the download operation is completed, the Target is automatically switched to Operation mode and the downloaded project is started.

Any time a project is changed, the modified files needs to be transferred to the Target device. When updating a Target, JMobile Studio provides the option **Download only changes** to transfer only the modified files to the device. The figure below shows the **Advanced** options expanded.

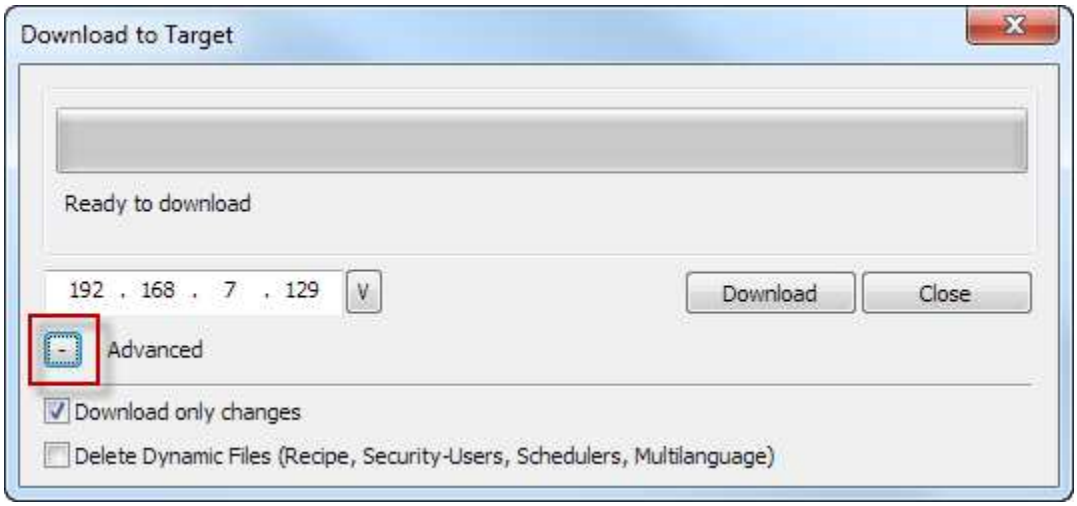


Figure 42

The other option is **Delete Dynamic Files**. There are files that can be modified in the HMI at runtime, for example you can create new users at runtime or you can upload new values to the recipes. If the option to delete the files is selected, the edited configuration of the recipes, or users, or the schedulers will be deleted and overwritten by the project configuration.

NOTE *Dynamic files are not deleted using* **Delete Dynamic Files** if storage type is external (example USB / SD Cards).

IMPORTANT Please make sure to check carefully before selecting this option as some data can be lost. Data cannot be restored after it has been deleted.

When transferring a project, the Studio uses a combination of HTTP and FTP connection. The HTTP connection is used to issue commands to the target device like “switch to transfer mode” or “unload running project”; the FTP session is instead used to transfer the files to the panel flash memory.

The Default port for HTTP connections on the Target is set to 80. However, the user can change the port number to a different value. To set the port number from JMobile Studio, click on the **Run > Manage Target**, and then click on **Target Setup on the dialog.** The HTTP, FTP port or HTTPS, FTPS port can be set for the target.

The Host Name can be defined by the user, in the appropriate box in the Target Port pop-up. This will allow each panel to be easily identified on a network with multiple panels. The drop down box will no longer show HMI@10.2.0.6, but will show, for example, Machine1@10.2.0.6.
After renaming the host, it is necessary to download the system files to the target.

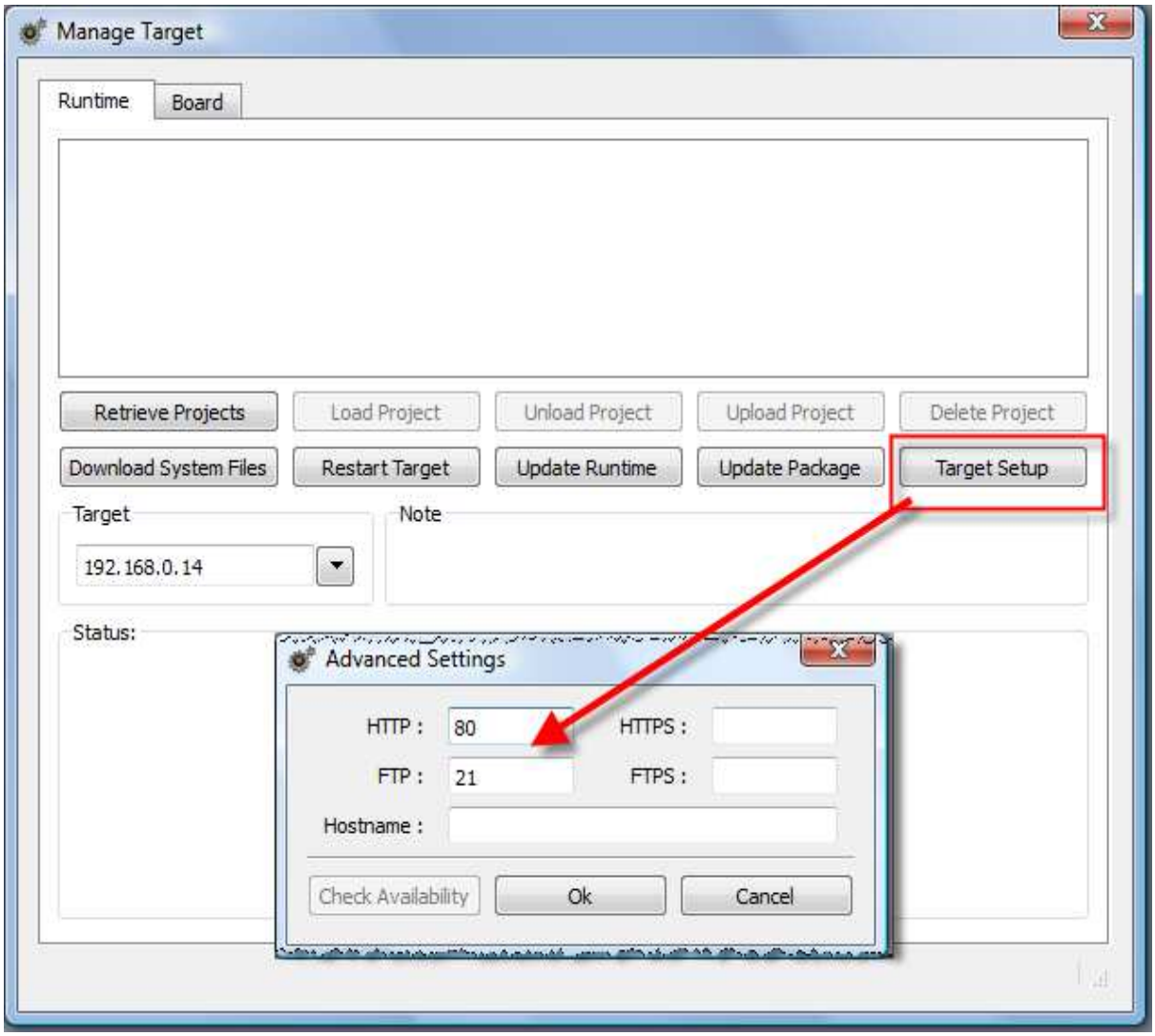


Figure 43

NOTE *Transferring a project after the above settings will result in a change of the default configuration. At the next download, the new ports will be used on the target and the new ports will have to be specified in the software to match the new selection.*

In the download dialog, click on Advanced Menu and set the port.

Figure 44

Set the HTTP/HTTPS port and FTP/FTPS port of the Target. They represent the port numbers the software uses to connect to the FTP(S) and the HTTP(S) servers on the Target. This is useful whenever default ports are, for some reason, in use by other applications or services, or if the local network requires using different port settings.

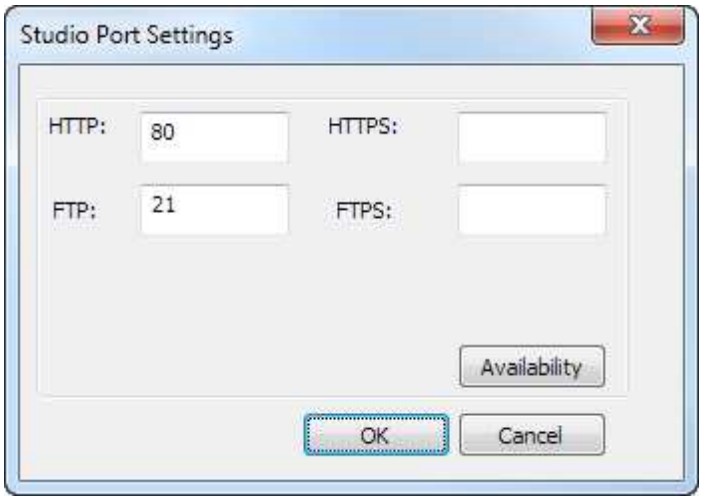


Figure 45

When Target Flash Memory is Low

While trying to download a project to the Target, if the project size is almost near or greater than the free space available in the flash memory, then it is not possible to download the project directly. The difference between the project size and available free memory should be at least 2MB.

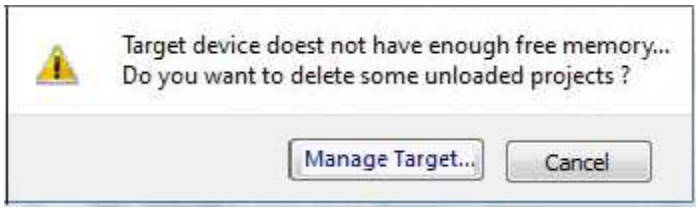


Figure 46

While clicking Download, a warning message will pop up mentioning that the Target memory is low and whether you need to delete some projects (as in the figure given above). Soon after you click “Manage Target”, the Manage Target window will open showing all the available projects in the Target. Deleting the unwanted projects from the target creates more memory space, hence making it possible to download the current project. By pressing Cancel, the dialog will close, and the download operation is aborted.

NOTE *The automatic check for available space for project download is a feature present from JMobile Studio and runtime version 1.80.*

5.2 Update Package

Both Runtime and project can be installed or updated using an update package via USB. To create an update package proceed as follows:

1) From the Run menu in the top toolbar, select **Update package**
2) Select the target and components you need to update

3) Specify the output directory for update package (example. USB flash drive)
4) Click **Create** to generate package
5) Assuming you have stored the package in the root folder of a USB drive, remove the drive from the PC, plug it into the HMI, activate the context menu by holding your finger for a few seconds on the screen (see also "Basic Unit Settings") and select "Update…" as shown in the figure below.

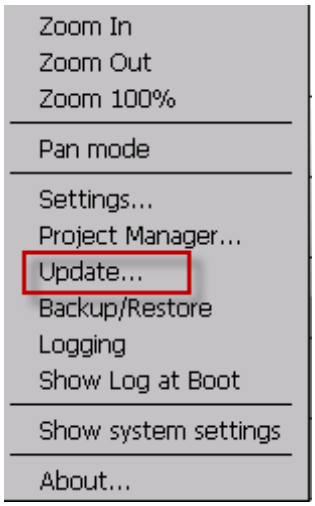


Figure 47

6) The system will automatically check for the presence of the update package in the root of the USB drive and ask confirmation to proceed with the update according to the figure below.

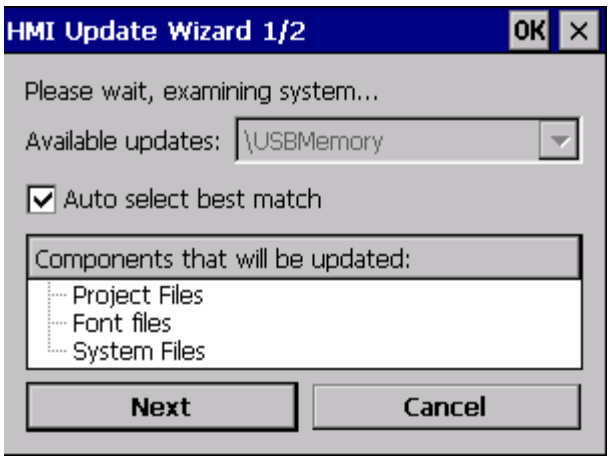


Figure 48

7) Mark the "Auto select best match" check box and click the "Next" button. The rest is automatically done by the system.

IMPORTANT *It is always recommended to create update packages with both flags* ***Project*** *and* **HMI Runtime** *checked. Use latest runtime with old project not converted with* JMobile Studio can originate stability problems.

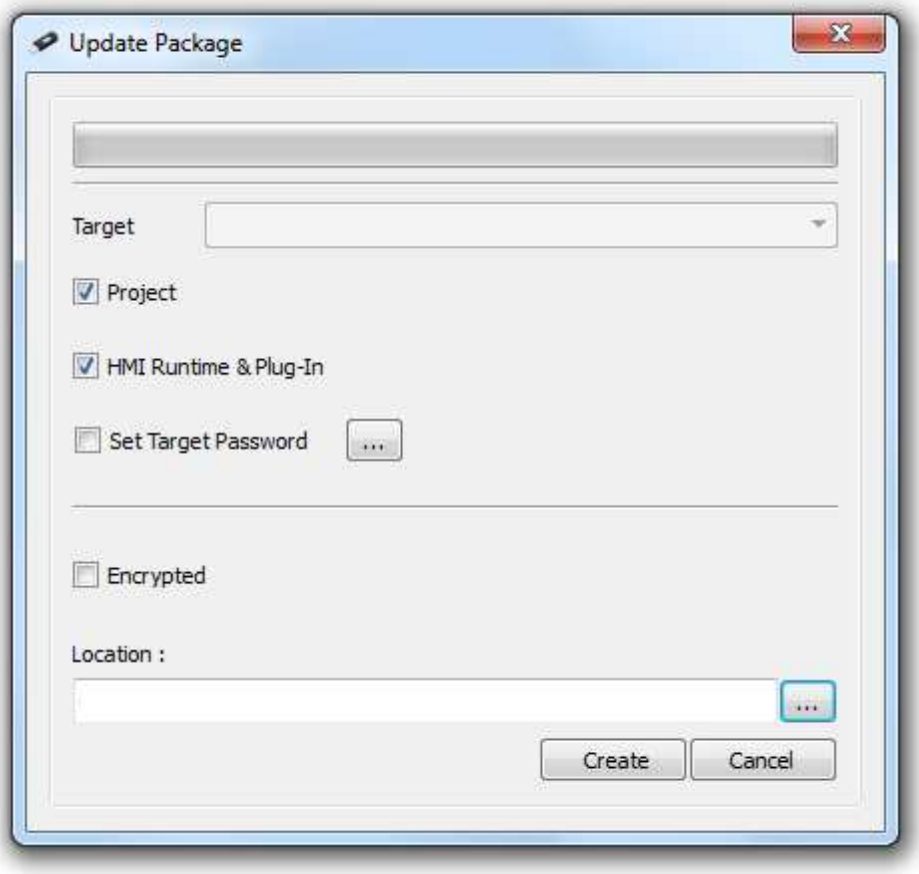


Figure 49

Target	Target type. When a project is open the target type is selected automatically otherwise it is responsibility of the user to select the correct target type.
Project	Project opened in *JMobile Studio is added to the update package.*
HMI Runtime & Plug-In	HMI Runtime is added to the update package. If a project is open in *JMobile Studio*, also required plugins will be added to update package.
Set Target Password	Can be used to set password used by HMI Runtime to protect operations like upload of projects, board management, download of projects, etc. Ref. to Remote access protection to HMI Panels for more information on to access protection.
Encrypted	Enable Encryption of update package; it cannot be read by any user and can be unzipped only by the HMI Runtime.
Location	Path for saving the update package.

5.3 The Runtime Loader

The explanations provided in the previous chapters are valid when using a panel with the runtime system already installed.
The HMI devices are delivered from the factory without the runtime. When you power up the unit for the first time, it starts with the "**Runtime Loader**" screen as shown below.

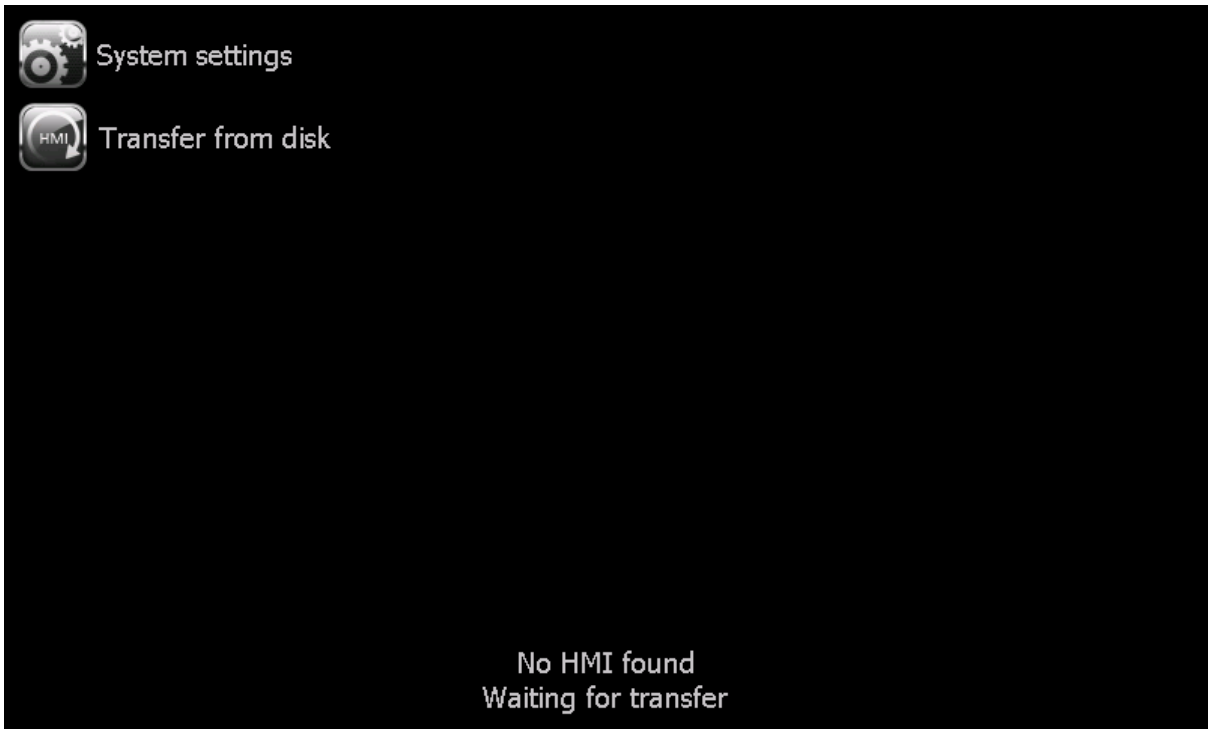


Figure 50

NOTE *The **Runtime Loader** is a feature dependent on the device Operating System and may not be available on all the units. The description provided in this chapter assumes that you are using JMobile Studio V1.80 or later. On MIPS based units, the Runtime Loader is available from version V2.65; on ARM based units the Runtime Loader is supported from BSP version V1.52.*

When you click on “System settings” you can activate the System menu in User mode, where you can set the IP address of the panel. See the chapter “System settings tool” for additional information on this tool.

Once the IP address is assigned and the panel is connected to a valid network, the easiest way to install the runtime is to download a project from the JMobile Studio. See the chapter “Transferring the Project to Target“ for additional information.

The normal download procedure in Studio is able to recognize the need for transferring the runtime and the process is automatically started. As soon as the panel IP is selected from the list of available units in the network, Studio will recognize the need for transferring the runtime, providing the information as shown in the following figure.

Figure 51

Just click on the Install runtime button to proceed.
The process will automatically go through the required steps, ending with the project download.

On an off-the-shelf unit the runtime can be installed also using an USB pen drive.

Prepare the Update Package according to the instructions provided in the chapter "Transferring the Project to Target" and make sure to mark all the check boxes for the HMI Runtime as shown in the following figure.

Then plug the USB drive in the panel and click on the "Transfer from disk" button.

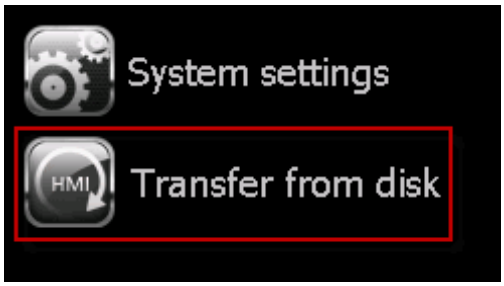


Figure 52

Then follow then the instructions on the screen.

NOTE *The Runtime Loader on the panel does not support the automatic installation of the runtime with versions prior to 1.80; in case an older version of the runtime has to be used on a unit with the Runtime loader, please contact technical support for additional information.*

5.4 Upload Projects

You can retrieve a project from a target device using the command "Upload Project". A copy of the project is transferred from runtime to the computer running *JMobile Studio*.

To upload a project proceed as follows:

1. Run -> Manage Target
2. In tab "Runtime", Select IP of the device from "Target" menu.

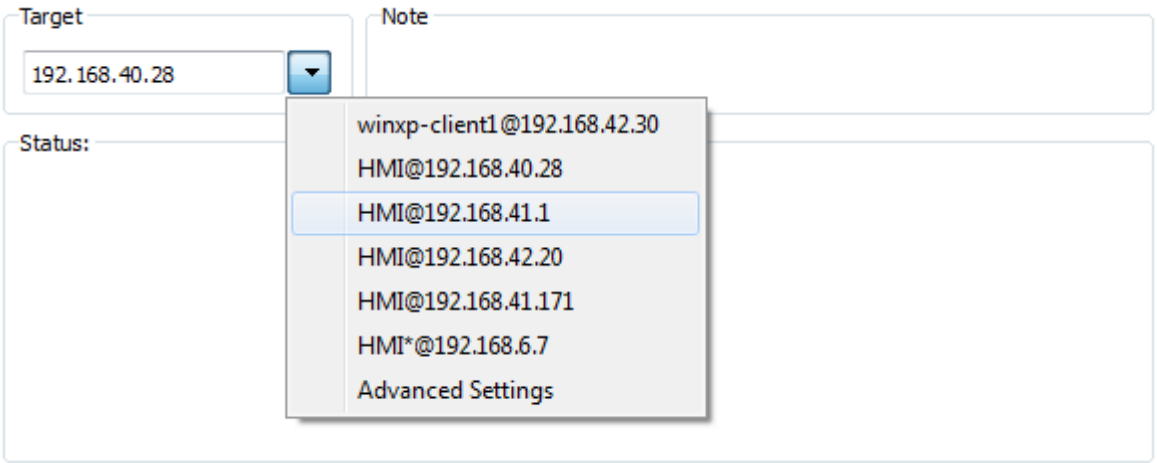


Figure 53

3. Click on “Retrieve Projects” to list all projects available in the target device
4. Select project to upload
5. Click on “Upload Project”
6. Enter password
7. Upload process starts

Once upload has completed, a copy of project is available in: *C:\Users\username\Documents\JMobile Studio\workspace\Uploaded\RuntimeIPAddress\workspace\ProjectName*

Starting from *JMobile Studio v1.90* (build 608) upload is no longer based on User Management for **access protection** but is protected by a dedicated password scheme. Please refer to Remote access protection to HMI Panels for more information related to access protection.

NOTE *If upload operation failed please verify firewall settings of PC where JMobile Studio is installed.*

6 Programming Concepts

The programming guidelines for JMobile Studio are based on a few basic concepts, which are common in many parts of the system.

6.1 Attach to

In JMobile Studio the basic programming techniques are used to configure the properties of an object placed on a page. Object properties can be set at programming time or configured to be dynamic.

To change a property at programming time you can use the page toolbar or the property pane which shows the properties available for the selected object.

Figure 54

The page toolbar permits a quick change of the most commonly used object properties.
When you need a complete view of all the properties of a certain object, you need to use the property pane. You have to select an object to see its properties shown in the property pane.
The property pane allows you to both change a property at programming time and attach the property to a dynamic element.
From the property pane, when you click on the right side of a property cell, you get the ability to "Attach to" the property to a tag. This operation is done using the "Attach to" dialog shown in the figure below.

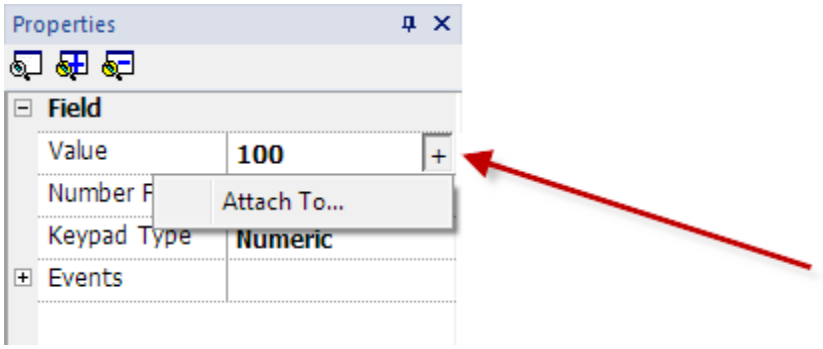


Figure 55

The “Attach to” dialog has two tabs. The first is called "Tag" and allows you to attach the property to an element. The "source" can be selected using the radio buttons.

The elements to which the property can be attached are:

- **Tags**
- **System** Variables (see chapter System Variables for an explanation of the meaning of all System Variables)
- properties from another **Widget**
- elements of a **Recipe**

The radio buttons at the bottom allow you to set the access type.
The TagIndex selection is used in the case of arrays to determine the array element.

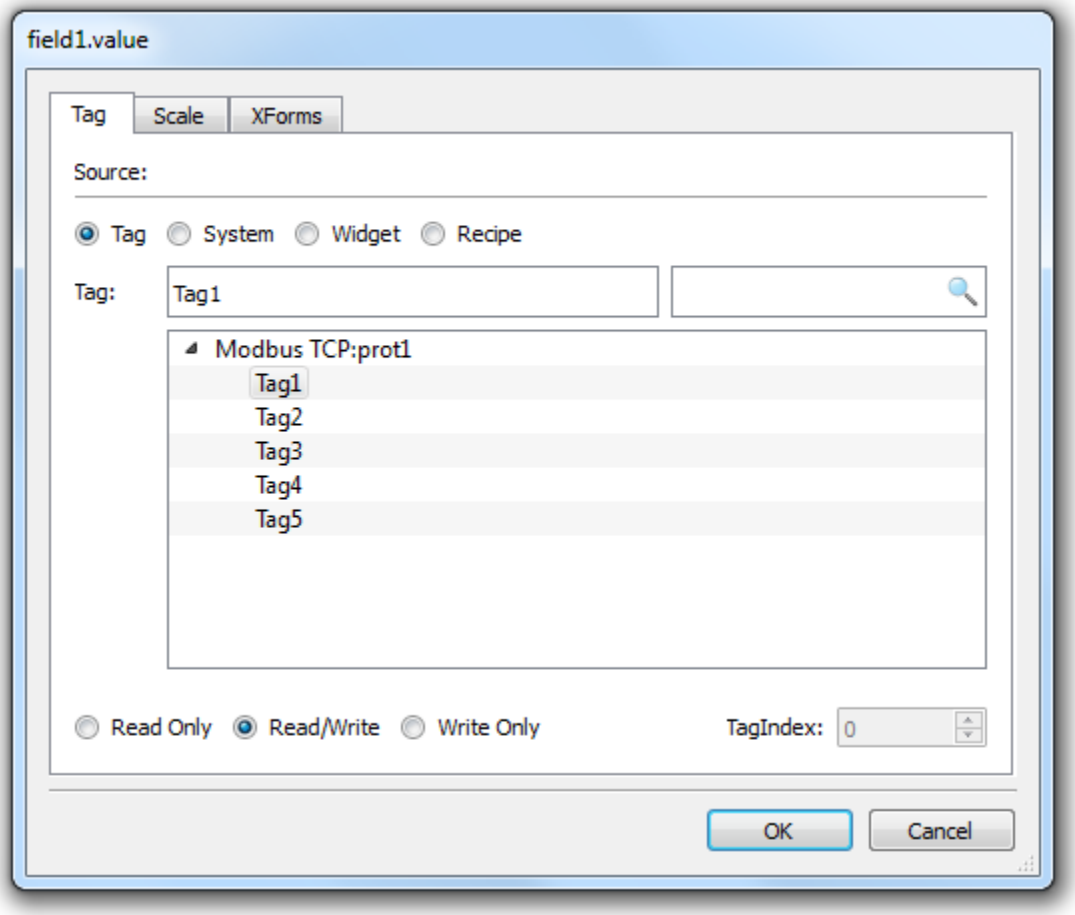


Figure 56

When adding Tags, the Protocols used in the Project are shown in the tag Dialog box and when expanding each protocol, the corresponding tags can be seen. The tags will be arranged in alphabetical order inside each Protocol.

There is an Option to search the tag to be attached by its name as shown in the Figure above.
This makes it easy to find Tags. The search can be done in two ways: first, you can start typing the tag name in the left box and this will “jump” into the list to the first tag starting with the characters you have entered; second, you can type in the search box any part of a tag name and this will automatically apply a filter to the view so that only the tags which contain the search characters are displayed.

Scale/XForms allows you to apply transformations to the numeric value of the source element before it is applied to the property of widget. Transformations can be simple linear relationships or generic transformations.
Linear scaling can be configured when selecting the "**Scale**" tab and they can be specified in terms of a formula or "**By range**". In case the range mode is selected, you just need to specify the input and output range and the system will automatically calculate the factors for the formula.

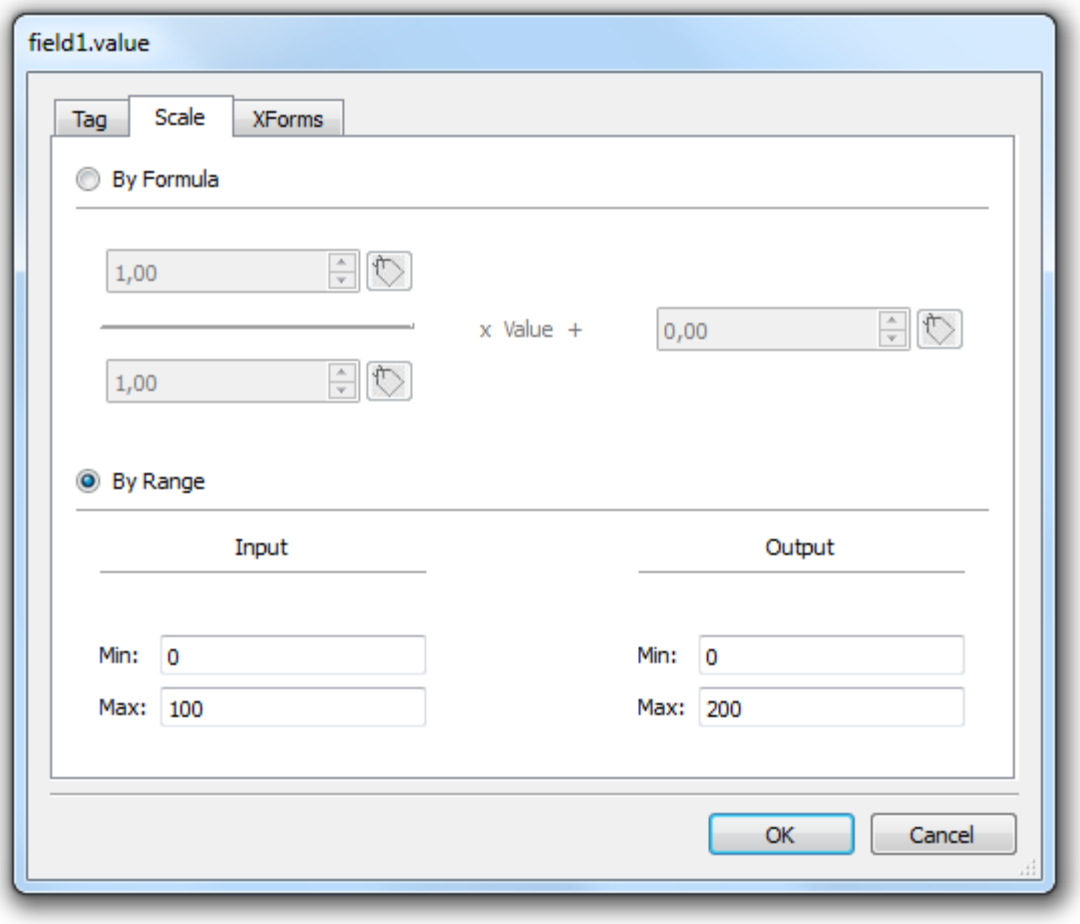


Figure 57

XForms transformations are applied to the result of scale transformation. Supported transformations are:

Color conversion Allows you to define a map between numeric values of the tag and colors to be assigned to the property. This feature is used to change the color of a button, for example, based on the value of a tag. If the tag is an integer, you can have many different colors based on the tag value

Bit and Byte index Allows extracting a single bit or byte content from a word depending on the specified bit or byte number.

Example of transformation: scaling (100/10*value + 5), byteIndex(0), bitIndex(1) , equivalent to: bitIndex (byteIndex (100/10*value+5,0) , 1)

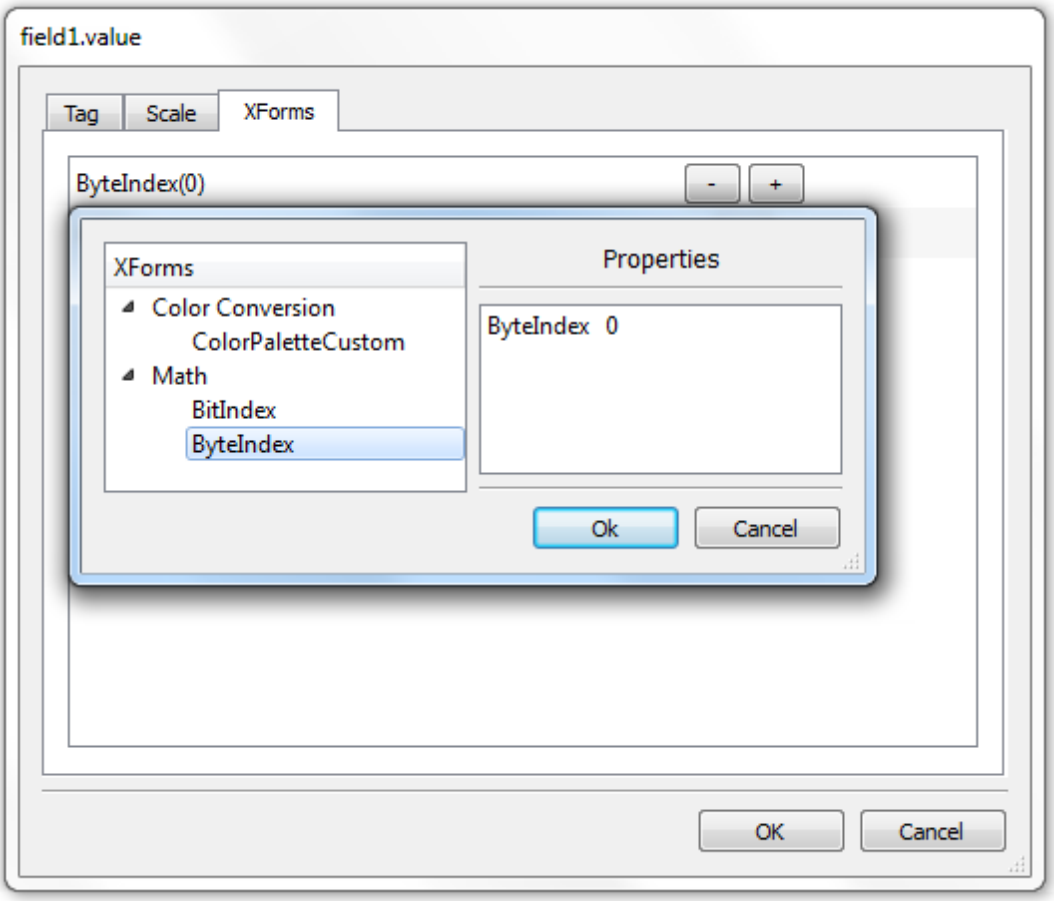


Figure 58

6.2 Events

In a JMobile Studio application, Events are the way to trigger Actions at the application level.

Main types of Events:

- Events related to buttons / touch (Click, Press, Release, Release)
- Events related to external input devices like keyboards & mouse (Click, Press, Hold, Release, Wheel)
- Events related to data changes (OnDataUpdate)
- Events related to switch of pages (OnActivate, OnDeactivate)
- Events related to alarms
- Events related to scheduler

Whenever the system generates an Event, you can attach one of the following actions to the event:

- an Action/Macro (or sequence of) selected from a list of predefined actions
- a JavaScript function

The figure below shows an example of an Action activated by pressing a button.

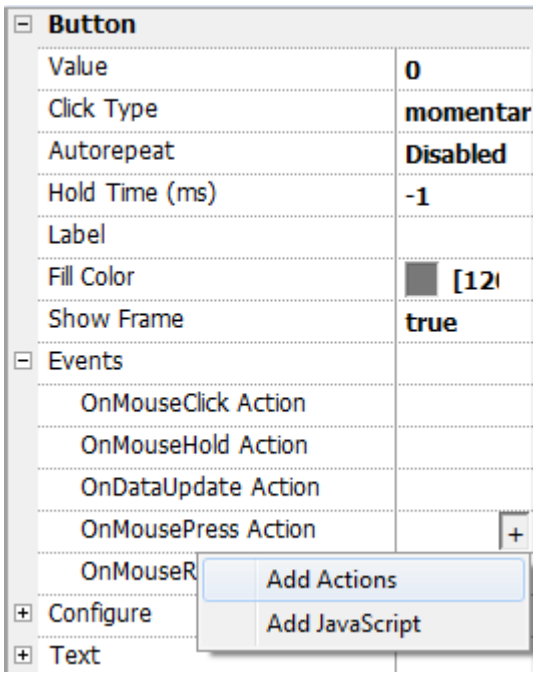


Figure 59

By associating Actions to Events, the programmer configures user interaction with the program.

6.2.1 OnClick / OnMouseClick

This Event occurs when the button/key is pressed and released quickly.

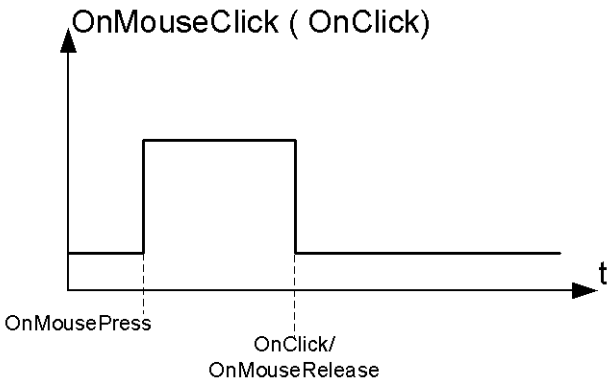


Figure 60

6.2.2 OnHold / OnMouseHold

This Event occurs when the button/key is pressed and held pressed for a certain **Hold time**. Actions programmed for this Event will be executed only after the Hold time has expired.

Default Hold time is configured in project properties but can be redefined for each button/key.
When a value **-1** is specified as Hold time for a certain button, the project default value will be used.

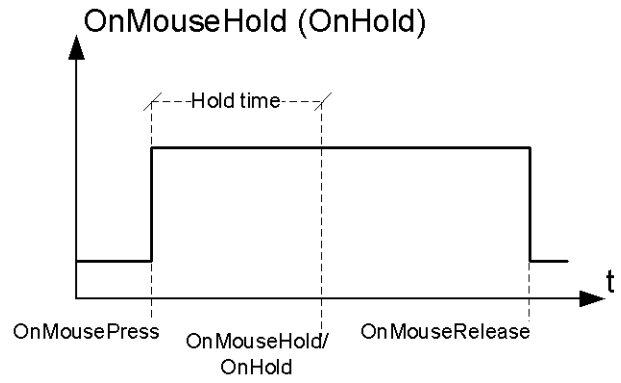


Figure 61

6.2.3 Autorepeat

It is possible to enable autorepeat for Press event or for Hold event of a button/key.
Autorepeat Time is specified in Project properties but can be redefined for each button/key.

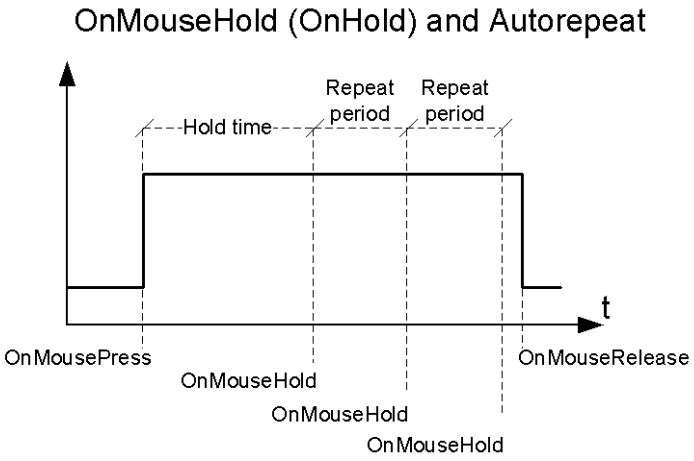


Figure 62

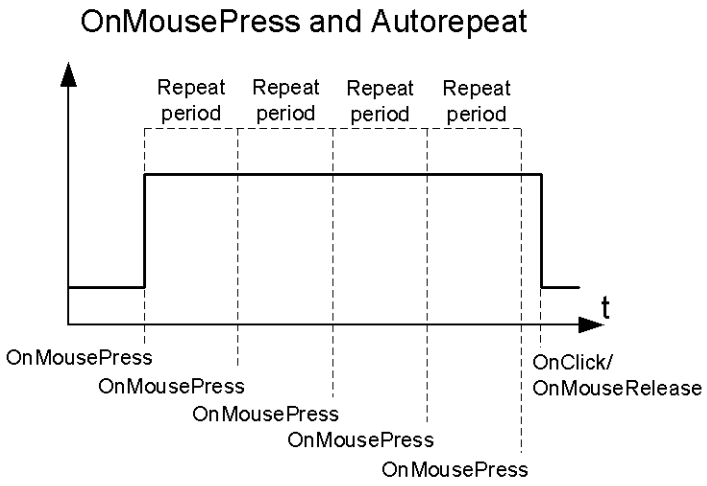


Figure 63

6.2.4 OnWheel

This Event occurs when a wheel (example: a USB mouse wheel or a handheld wheel) value change. A wheel usually is used to increment/decrement a value in data entry or attached to a tag.

6.2.5 OnActivate

This event triggers when a page is loaded and before widgets being initialized with the values read from Tag Manager.

6.2.6 OnDataUpdate

This event triggers when a data field attached to a widget changes. Upon page change, data is updated asynchronously at a time that depends on the time needed to read data from protocol. As a consequence, the OnDataUpdate event can be triggered or not, depending on whether data becomes available from protocol respectively after or before widgets being initialized for first time. In particular page change notifications are more likely to happen with slow protocols and remote clients.
Moreover, note that the value we read during OnActivate can be the same we get from a subsequent OnDataUpdate event, since OnDataUpdate notifications are sent asynchronously.

6.3 Widgets positioning: Snap to Grid / Snap to Objects

To help user in precise widgets positioning two editing options options are available in JMobile Studio:

- **Snap to Grid**
- **Snap to Object**

“Snap to” positioning can be enabled via the top toolbar →**View** menu items.

6.3.1 Snap to Grid

In **Snap to Grid** mode when you move or resize an object, its top left corner will align or "snap to" the nearest intersection of lines in the grid, even if the grid is not visible. If default settings does not fit your needs, you can either switch off the function of customize the grid.

The Grid can be enabled from **View →Show Grid** and customized from **View→Properties** where it is possible to choose:

- **Spacing X**: space in pixel between two lines/dots in X axis
- **Spacing Y**: space in pixel between two lines/dots in Y axis
- **Type**: type of grid, **dot** or **line**
- **Color**: the color of grid

6.3.2 Snap to Object

In **Snap to Object** mode when you move an object, it will align or "snap to" other objects.

When you select an object to move, one of the following hot points is selected as the source of the snap point, depending on the area you pressed: top, top left, top right, bottom, bottom left, bottom right, left, right, center:

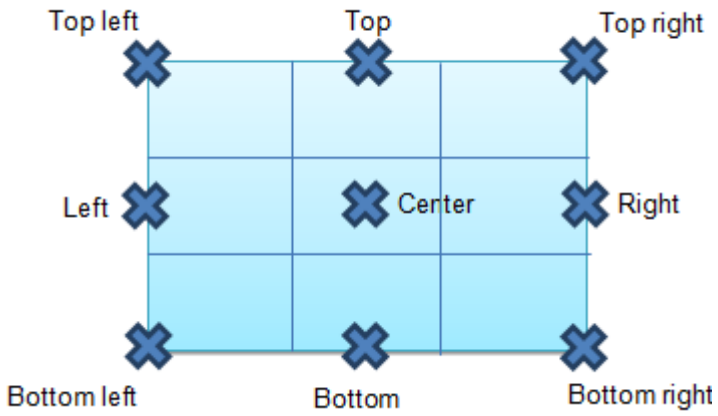


Figure 64

The algorithm tries to find a matching hot point among the neighbourhood widgets hot points which matches either x or y coordinates of the source snap point.

For line widget the source snap points are the terminal points of the line.

6.4 Z-order of widgets

Z-order is an ordering of overlapping two-dimensional objects / widgets such as shapes (or objects in a 3D application). One of the features of widgets is that they could overlap, so that one widget hides part or all of another. When two widgets overlap, their z-order determines which one appears on top of the other.

A widget with greater z-order is always in front of an element with a lower z-order.

Z-ordering of widgets is essential for performances since overlapping dynamic widgets can invalidate *static optimization* and reduce performance of hmi applications.

In JMobile Studio a new toolbar is available to help user understand widgets overlapping. The toolbar allows to:

- Enable visual filtering: hide widgets stacked above and/or below selected widgets (using the two buttons available in toolbar).
- Check z-order of widgets using the combobox. An icon allows to distinguish between static (picture icon) and dynamic (movie frame icon) widgets. The combobox is listing widgets in z-order.

Visual filtering reduces the opacity of the widget above the currently selected widget, so user can easily edit widgets hidden by overlapping items.

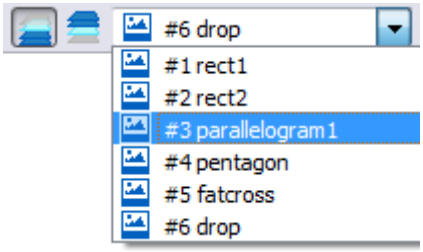


Figure 65

6.5 Change properties of several widgets at once

For widgets of the same type (ex. two or more labels, two or more fields, two or more gauges etc) the user can change common properties in few steps using JMobile Studio.

To change multiple properties of widgets proceed as follow:

1) Select all widgets of the same type where is required to change common properties
2) Change common properties from property pane

When multiple widgets are selected, the property pane title changes to "**<MultipleObjects>**" indicating that all properties changes will be applied to all widgets selected on the page.

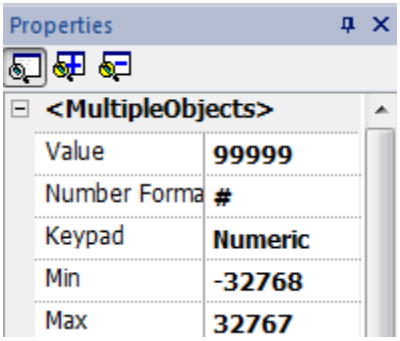


Figure 66

NOTE *Not all properties can be modified for multiple widgets simultaneously and must therefore be modified individually.*

7 Project Properties / Project Widget

Project properties contain settings for the project. Project properties are available from **Project View**.
The **Properties** window on right side of the JMobile Studio contains the list of project level user-configurable data.

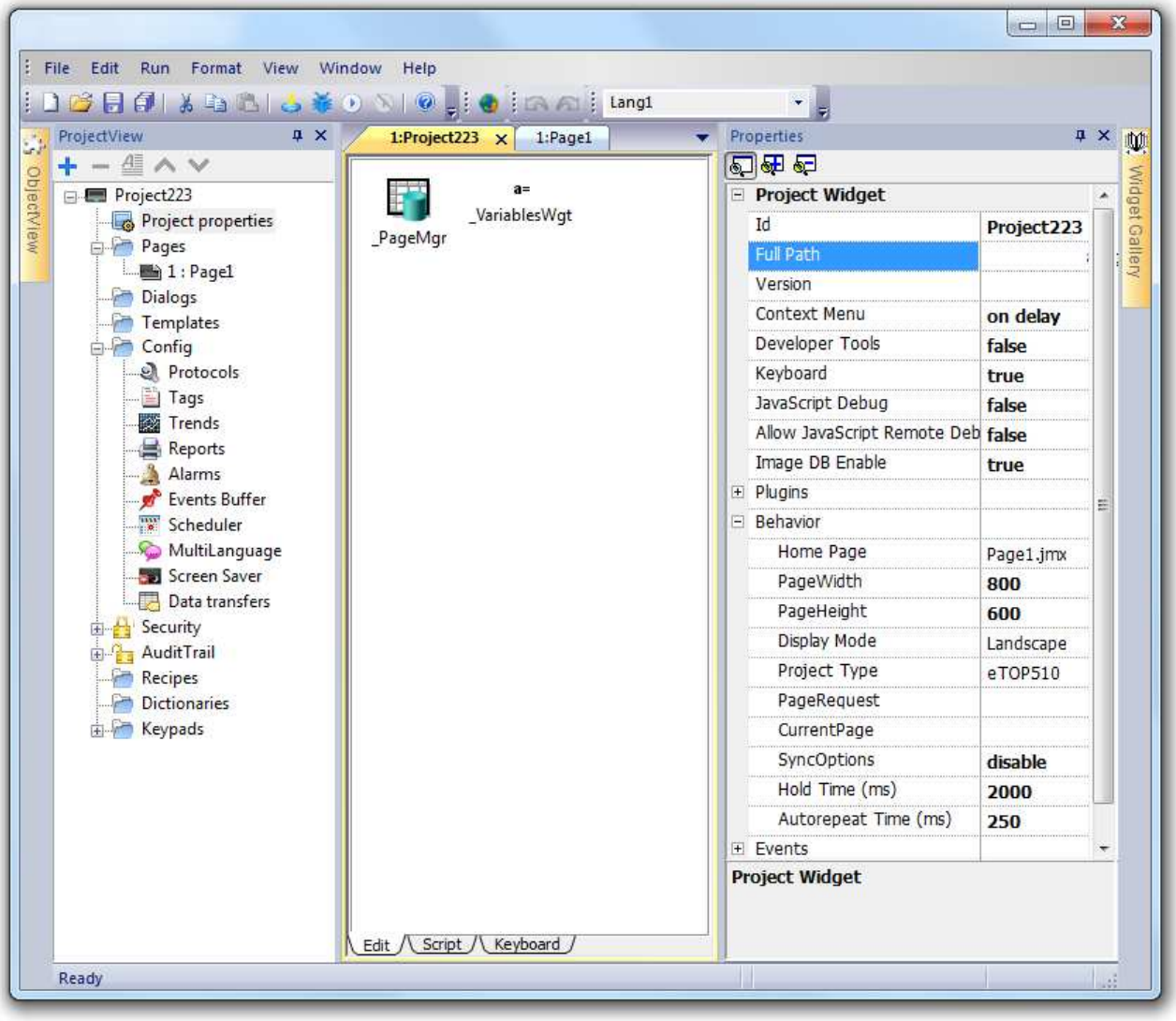


Figure 67

7.1 Version

The Version field is available for users to report the project version.

7.2 Context Menu

The default method for users to access the runtime settings is to press and hold for a few seconds on an empty area of the runtime screen. Using this property you can choose how the context menu should appear:

on delay (default) press and hold
on macro command via macro/action controlled by HMI application

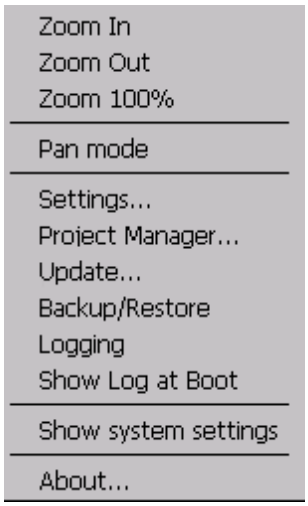


Figure 68

7.3 Developer Tools

Developer tools are a collection of utilities useful for debugging problems at runtime. To use developer tools proceed as follow:

1) set to true **Project properties ->Developer tools** in JMobile Studio
2) Download the project to the target
3) Open context menu
4) Select **Developer tools**

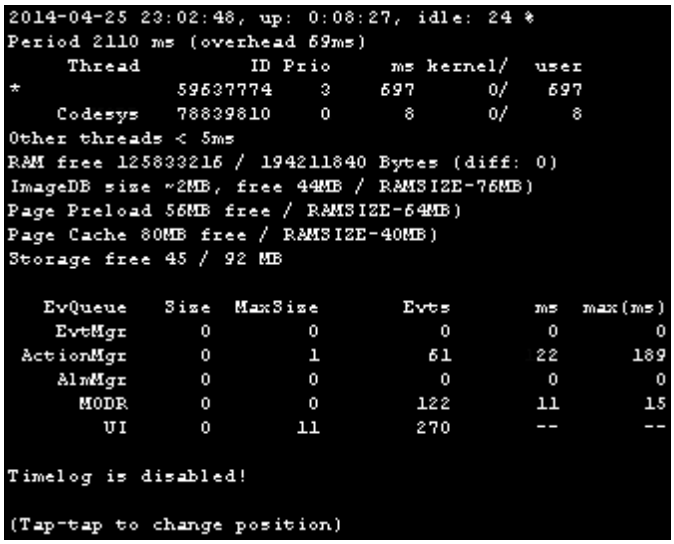

2014-04-25 23:02:48, up: 0:08:27, idle: 24 %
Period 2110 ms (overhead 69ms)
Thread ID Prio ms kernel/ user
* 59637774 3 697 0/ 697
Codesys 78839810 0 8 0/ 8
Other threads < 5ms
RAM free 125833216 / 194211840 Bytes (diff: 0)
ImageDB size ~2MB, free 44MB / RAMSIZE-76MB)
Page Preload 56MB free / RAMSIZE-64MB)
Page Cache 80MB free / RAMSIZE-40MB)
Storage free 45 / 92 MB
EvQueue Size MaxSize Evts ms max(ms)
EvtMgr 0 0 0 0 0
ActionMgr 0 1 61 22 189
AlmMgr 0 0 0 0 0
MODR 0 0 122 11 15
UI 0 11 270 -- --
Timelog is disabled!
(Tap-tap to change position)

Figure 69

The list of items available in developer tools is reported below.

Show/Hide all	Show an overview of all main information about device status like CPU Load, memory usage, event queues etc. in one dialog in overlay.
Cpu statistics	Show information about CPU load. On top the actual machine time is shown along with the total machine uptime. CPU statistics are collected with a frequency of 2000 milliseconds. The actual period time and the overhead required to collect and visualize statistics are displayed as well. The more the actual period time is far from the nominal 2000 milliseconds the more the system is busy. Cpu consumption of threads ofY is listed reporting the name of the thread (if available, main thread is marked with a *), the thread ID, the thread priority and cpu time spent during the 2sec period, divided in user and kernel time.
Memory statistics	Show information about CPU load. In particular the free system RAM is shown along with the difference of memory usage from previous iteration (a negative value means free memory is decreasing).
Event queues	Show information about the size of event queues. Many core components of the HMI are event driven. For each event queue the actual size of the queue, the maximum achieved size of the queue, the total number of events processed and the last and maximum times required to process events are shown respectively (timing statistics are only available for non-UI queues).
Timelog summary	Show information about time spent for loading active page. See 7.3.1 for more details.
Embed window	Allows to embed in runtime the scene or leave the developer tool window as a standalone window (dialog).
Reset queue stats	Reset information collected related to event queues.
Disable watchdog	Disable the watchdog and avoid system restart in case of freeze or crash of services
Ignore exceptions	Usually when an exception is captured the crash report panel is shown. By setting this option exceptions will be ignored and no crash report window appears.
Launch VNC	Launch VNC server if available in runtime. VNC server is available as a plugin for WCE target only.
Profiling	Using this option it is possible to check the time spent for loading/rendering the active page.

7.3.1 Profiling

Using this option of the developer tools, it is possible to check time spent for loading/rendering the active page. Profiling will be available from the next page load and only for the first painting of the page to the screen (please note that the configuration is retained).

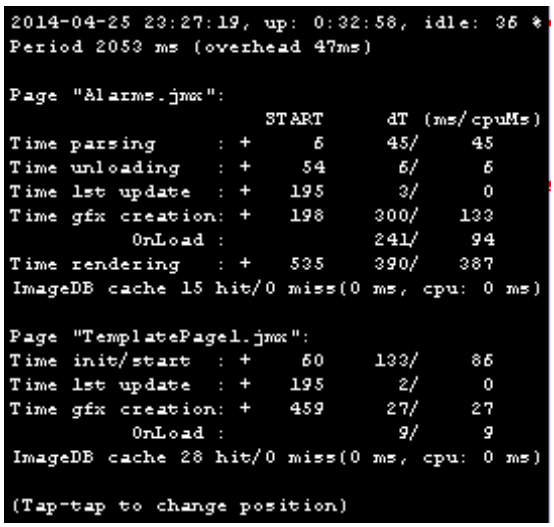

2014-04-25 23:27:19, up: 0:32:58, idle: 36 %
Period 2053 ms (overhead 47ms)
Page "Alarms.jmx":
START dT (ms/cpuMs)
Time parsing : + 6 45/ 45
Time unloading : + 54 6/ 6
Time lst update : + 195 3/ 0
Time gfx creation: + 198 300/ 133
OnLoad : 241/ 94
Time rendering : + 535 390/ 387
ImageDB cache 15 hit/0 miss(0 ms, cpu: 0 ms)
Page "TemplatePage1.jmx":
Time init/start : + 60 133/ 86
Time lst update : + 195 2/ 0
Time gfx creation: + 459 27/ 27
OnLoad : 9/ 9
ImageDB cache 28 hit/0 miss(0 ms, cpu: 0 ms)
(Tap-tap to change position)

Figure 70

Profiling data can be:

- saved to file (**Save timelog to file**): save a .txt report of time spent loading project & pages. Usually is better to avoid it since it could have high and unpredictable effects on page change performance. However it's useful to export & share profiling details.

- visualized using context menu (**Timelog summary**): a summary about current page (and related template if any). There you can see partial times for different operations. Usually the most important ones are:
    - **Time parsing:** time spent for parsing current page file / .jmx. Parsing time usually is proportional to the number of widgets in a page (so to the complexity of a page).
    - **Time gfx creation**: which is mainly contributed by the OnLoad methods, where typically images are loaded.
    - **Time rendering**: time spent for rendering the scene
    - **Time unloading**: time spent for unloading page(in case the current page comes from another page).

    Times are provided in couples: wall time/cpu time. Wall time is the absolute time required by this part which can be higher than the actual cpu time used to do it because we can have higher priority threads running (for instance protocols). There is also a start time column which is relative to page load start time. It can be useful to track the actual time required to load a page, since partial ones only focus on the most time critical functions missing the rest that often contributes significantly to the total time. For example the actual total wall time required to load a page is rendering (which is the last step) start time + rendering wall time.

- overlayed as colored rectangles on the view (**Overlay OnLoad/Rendering times**):this view allows to effectively represent time spent on single widgets and is available only for the rendering and OnLoad steps. The view gives an immediate feeling of where time is spent. Red zones represent the most time critical zones. Detailed widget times are visualized by a tooltip window (on Win32 platform attached to mouse hover event, on WCE press drag and release over the region of interest). In case of out-of-the-scene widgets some arrows allow to navigate to these areas and hovering on them the tooltip will show the area summary.

7.3.2 Watchdog

One of the most important items in the Developer tools is related to the **watchdog**. This item allows the developer to disable the watchdog to avoid a system restart in case of a runtime crash, to have time to save the **crash report** or check system status information (example: memory available, CPU load, events queue size etc).

Crash report dialog appears automatically in case of a system freeze or crash and allows users to save a log file of crash. The crash report may contain information important for technical support.

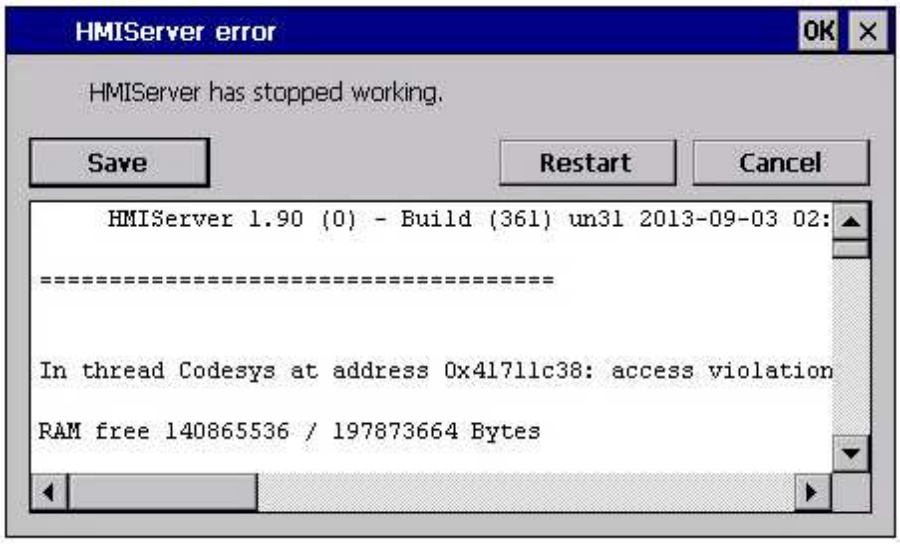


Figure 71

7.4 Buzzer on touch / Buzzer duration

Using this feature the runtime buzzes when a user presses one of the following widgets:

- Buttons
- HotSpots
- Needles
- Fields
- External keys
- Combobox
- Tables items
- Control list items

Using **Project properties** ->**Buzzer duration** defines the duration of the buzz when an event is fired. Buzzer duration is a value in ms, default is 200ms.

NOTE Buzzer on touch requires *WCE v1.76 ARM / 2.79 MIPS and can be used as an alternative to the **Touch buzzer** feature available WCE side that buzz when user touch any point into the touchscreen. **Buzzer on touch** is supported also by Win32 runtime.*

7.5 Keyboard

Enable the use of keyboard Macros at runtime when using external keyboards.

7.6 JavaScript Debug

Enable the JavaScript debugger at runtime for current project.

NOTE *For UN20 target (WCE MIPS hmi panels), local debugger has been disabled. However, remote debugger is available to debug JS from a PC connected to HMI panel via Ethernet.*

NOTE Remote debugger not supported in JMobile Client and ActiveX.

7.7 Allow JS Remote Debugger

Enable the JavaScript remote debugger for current project.

NOTE *For UN20 target (WCE MIPS hmi panels), local debugger has been disabled. However, remote debugger is available to debug JS from a PC connected to HMI panel via Ethernet.*

NOTE Remote debugger not supported in JMobile Client and ActiveX.

7.8 Image DB enable

Enabled by default, this property activates an engine used by the runtime to optimize project performance. Available in the Project Properties, should be disabled just by tech support for debugging in case of a problem. Disabling it can create performance problems at runtime.

7.9 FreeType Font Rendering

The “FreeType Font Rendering” property is used to switch between old font engine used by JMobile Studio & Runtime up to v1.80 (native OS-based font engine) and the font rendering based on FreeType.
All projects created with JMobile Studio v1.90 (b608) or newer use the FreeType font engine as default while all projects created with older versions of JMobile Studio continue to use old font engine after the conversion to avoid potential backward compatibility issues in font rendering.
Moving to the FreeType Font Rendering is recommended to all users; to enable it set true in “FreeType Font Rendering” in Project Properties, save and verify that all texts are shown correctly in all HMI project pages.

Example of rendering issues that could appear when switching between old and new font engine are:

- text require few more/less pixels for rendering and this could change text layout
- size to fit could result in change in size of widgets.
- better rendering using antialiasing (feature not available in v1.80). Antialiasing can be disabled in v1.90 for texts (it is a property of text widgets).

7.10 Software Plug-in Modules

The Software Plug-in concept allows users to choose if certain software modules must be downloaded to the runtime together with the project. Example of Software Plug-in are:

- WebKit (module required by browser widget – if available).
- PDF Reader
- VNC Server
- ActiveX

Not all Software Plug-in modules are compatible with all targets. New software plug-in modules will be added in the future to extend optional features of the product.

Once enabled, a Software Plug-in is considered as part of the runtime. You can use JMobile Studio to install it in the target using one of the following procedures:

- Installing runtime / Updating runtime
- Update Package

Plug-ins can be removed once installed using one of the following operations from **System Mode**:

- Format Flash
- Restore Factory Settings

The system is not able to detect automatically if any Software Plug-in is required by the HMI application, so it is up to the user to select the Software Plug-in manually from the project properties when required.

Software Plug-in support has been designed for embedded HMI panels where storage is limited and reducing software footprint is critical. This option is not supported in Win32 platform.

7.11 Behaviour -> Home Page

Define Homepage of project. The homepage is the first page loaded at runtime (after log-in page if security is enabled in project).

When **Security** is enabled, it is possible to specify a different homepage for each groups of users, in this case this property is ignored. Refer to User Management for more details.

7.11.1 Behavior -> Page Width / Page Height

Define default size in pixel of an HMI page. Default is target type dependent (depend on HMI panel screen resolution).

7.11.2 Behavior -> Display Mode

Define HMI panel orientation, Landscape or Portrait.

7.11.3 Behavior -> Project Type

Define target type / HMI panel model. Based on model, many features and properties of project are automatically adjusted to fit it in the right way.

7.11.4 Behavior -> PageRequest, CurrentPage and SyncOptions

The HMI projects contain properties that let you know which page is currently displayed on the HMI and to force the HMI to switch to a specific page. These properties can be used to synchronize pages showed on the HMI and JMobile Client or to control an HMI with a PLC.

Double click on project name present into ProjectView pane to open the project properties page:

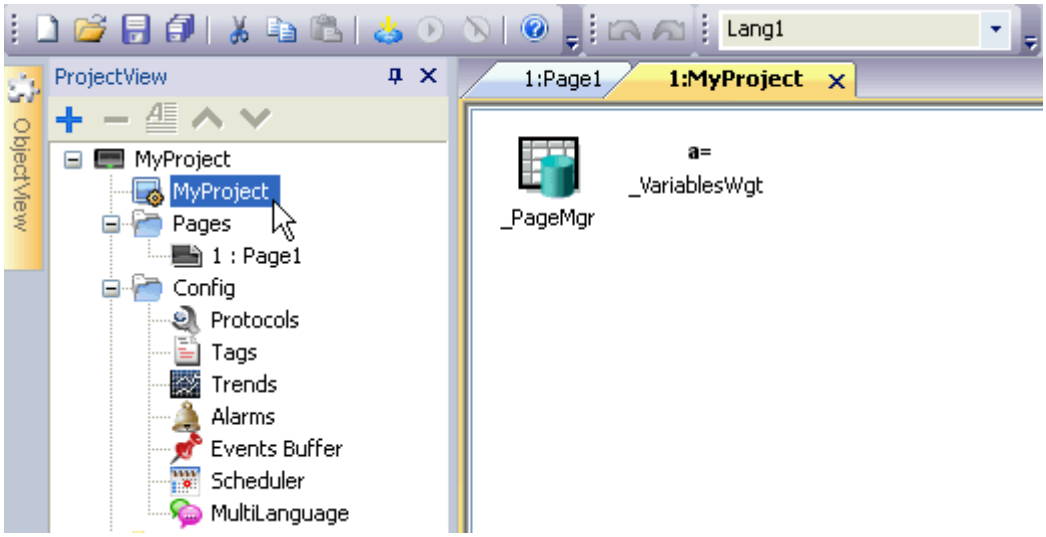


Figure 72

Expand the properties view of the Properties pane, by clicking on "Show Advanced Properties" button:

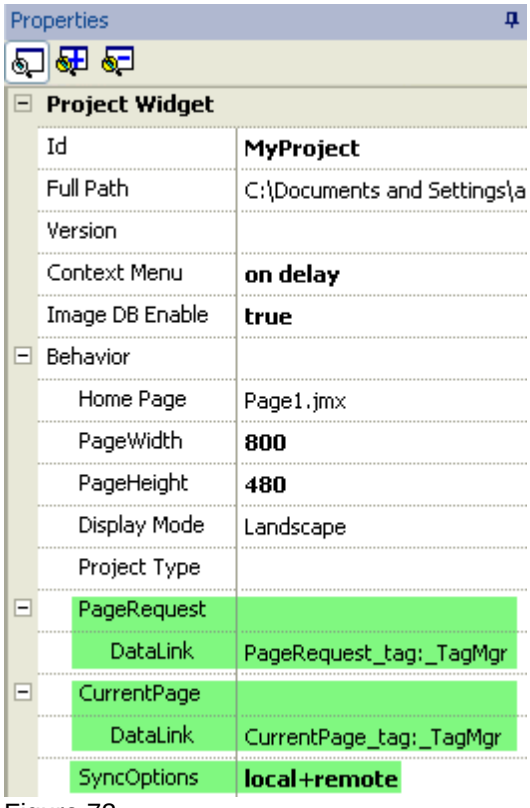


Figure 73

Following properties, highlighted in green in the picture above, can be configured:

PageRequest This property determines the page to be shown on the HMI and on JMobile Client.
Attached Tag must contain an integer value within the range of the available project pages.
The attached Tag must be available at least as a Read resource.

CurrentPage This property represents the page number actually displayed on the HMI or on JMobile Client or on both.
Attached Tag must be available at least as a Write resource and must have data type that allows containing an integer value.

SyncOptions value can be set as one of following options:

- **Local**: if you want that CurrentPage represents the number of page actually displayed on HMI,
- **Remote**: if you want that CurrentPage represents the number of page actually displayed on JMobile Client.

SyncOptions This property determines the synchronization of the project pages with the value contained into the CurrentPage property.

- **Disable**: CurrentPage value is ignored.
- **Local**: CurrentPage value corresponds to the page displayed on HMI.
- **Remote**: CurrentPage value corresponds to the page displayed on JMobile Client.
- **Local + Remote**: CurrentPage is changed according to page displayed on HMI and on JMobile Client, if different pages are displayed; CurrentPage refers to the last page loaded.

Examples related to the use of PageRequest and CurrentPage.

Example 1
Force page change from PLC to HMI and JMobile Client.

PageRequest	attached to Tag "A"
CurrentPage	Empty
SyncOptions	Disabled

Changing value of "A", HMI and JMobile Client will show page requested.

Example 2
Force page change from PLC to HMI and JMobile Client. Read current page loaded on HMI.

PageRequest	attached to Tag "A"
CurrentPage	attached to a Tag "B" as Read/Write
SyncOptions	Local

Changing value of "A", HMI and JMobile Client will show page requested. On "B" will be written page currently showed by HMI.

Example 3
Force page change from PLC to HMI and JMobile Client. Read current page loaded on JMobile Client.

PageRequest attached to Tag "A"

CurrentPage attached to a Tag "B" as Read/Write

SyncOptions Remote

Changing value of "A", HMI and JMobile Client will show page requested.
On "B" will be written page currently showed by JMobile Client.

Example 4
Force page change from PLC to HMI and JMobile Client. JMobile Client page Synchronization with HMI (not vice versa) .

PageRequest attached to a Tag "A" as Read/Write

CurrentPage attached to the same Tag "A" as per PageRequest

SyncOptions Local

Changing value of "A", HMI and JMobile Client will show page requested. Changing page on HMI same page will be forced on JMobile Client.

Example 5
Force page change from PLC to HMI and JMobile Client. HMI page Synchronization with JMobile Client (not vice-versa).

PageRequest attached to a Tag "A" as Read/Write

CurrentPage attached to the same Tag "A" as per PageRequest

SyncOptions Remote

Changing value of "A", HMI and JMobile Client will show page requested. Changing page on JMobile Client same page will be forced on HMI.

Example 6
Synchronize displayed page between HMI and on JMobile Client.

PageRequest attached to a Tag "A" as Read/Write

CurrentPage attached to the same Tag "A" as per PageRequest

SyncOptions Local+Remote

Changing page on HMI, same page will be shown on JMobile Client and vice-versa.

7.11.5 Behavior -> Hold Time / Autorepeat Time

Define default values for hold time and autorepeat time for buttons and external keyboards. However, for each button/key, they can be redefined in related widget instance.

7.11.6 Events -> OnWheel

A wheel is used by special products like handhelds or USB mouse with a wheel input device. Usually a wheel is used to increase/decrease the value of a Tag without the need to use an external keyboard device.

From **Project Widget it** is possible to attach to a change of wheel status an action like StepTag to increase/decrease a tag value.

8 System Variables

System variables are special tags containing information about the runtime.
System variables are available in the Attach to dialog from the "Source" selection as shown in figure.

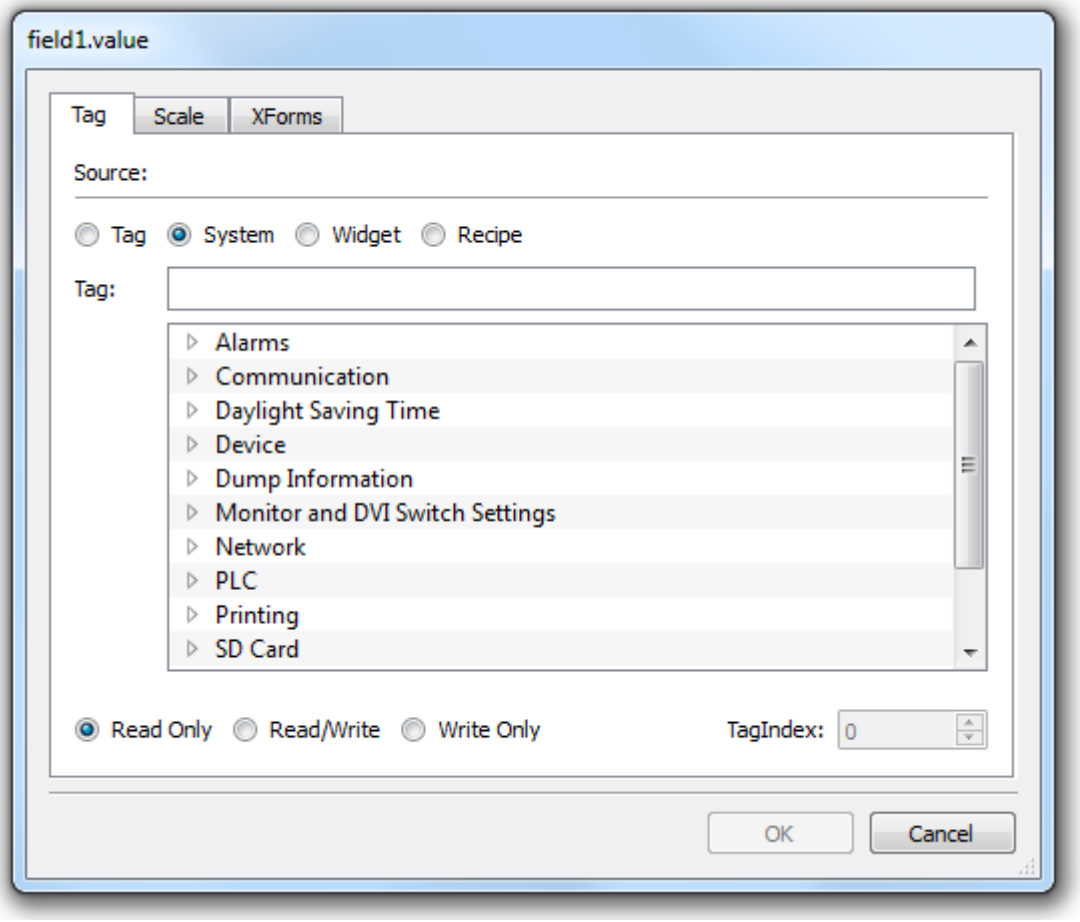


Figure 74

System variables are divided into categories.

Starting from v1.90 (b608), System Variables are available also as a standard protocol in the Protocol Editor. This protocol can be used for data transfer betweens System Variables and tags from devices, or when is necessary to specify a custom refresh rate for a System Variable.

8.1 Alarms

Variables return information on the actual number of alarms according to the status.

Not Triggered Acknowledged Total number of alarms **“**Not Triggered Acknowledged”

Not Triggered Not Acknowledged	Total number of alarms **“**Not Triggered Not Acknowledged”
Triggered Acknowledged	Total number of alarms **“**Triggered Acknowledged”
Triggered Not Acknowledged	Total number of alarms **“**Triggered Not Acknowledged”
Triggered Alarms	Total number of alarms “Triggered”
Number of missed alarm events	Total number of missed alarm events

All these System Variables are Int type (32 bit), read only.

8.2 Communication

Variables return information on the status of the communication between the HMI device and the controllers configured in the Protocol Editor.

Protocol Communication Status	The variable is read only Int (32 bit), and can have 3 values: **0 =** No protocol running; it may occur if the protocol driver has not been properly downloaded to the target system. **1** = Protocol has been properly loaded and started; no communication errors **2 =** At least one communication protocol is reporting an error
Protocol Error Message	This variable returns an ASCII string containing a description of the actual communication error. The communication protocol acronym is reported between square brackets to recognize the source of the error in case of multiple protocol configurations. The variable is a read only string. If no errors are present, the string will be blank.
Protocol Error Count	This variable returns the number of communication errors that occurred since the last time it was reset. The variable is a read only integer. The reset of this variable is only possible using the dedicated Action “Reset Protocol Error Count“

8.3 Daylight Saving Time

Variables return information on the system clock and allow adjusting it from the application. They contain information on the "local" time.

All the variables are read only; this means that you cannot change them to update the system RTC. All the variables are bytes (8 bit) except for the DLS and Standard Offset that are shorts (16 bit).

Standard time is the "solar time" and other is Daylight Savings Time.

Standard offset	Represents the offset in minutes when standard time is set, with respect to GMT. (with respect to the picture it is -8*60 = - 480 minutes)
Standard week	the week in which the Standard time starts (w.r.t. the picture it is First = 1)

Standard Month	the month in which the standard time starts (range of the variable is [0 - 11] so w.r.t. the picture it is November = 10)
Standard Day	day of week in which the standard time starts (w.r.t. the picture it is Sunday = 0)
Standard hour	hour in which the standard time starts (w.r.t. the picture in Time field it is 02 = 2)
Standard minute	minute in which the standard time starts (w.r.t. the picture in Time field it is 00 = 0)
Dst offset	Represents the offset in minutes when DLS time is set, with respect to GMT. (w.r.t. the picture it is -7*60 = - 420 minutes)
Dst week	Week in which the DLS time starts (w.r.t. the picture it is Second = 2).
Dst Month	month in which the DLS time starts (range of the variable is [0 -11] so w.r.t. the picture it is March = 2)
Dst Day	day of week in which the DLS time starts (w.r.t. the picture it is Sunday = 0)
Dst hour	hour in which the DLS time starts (w.r.t. the picture in Time field it is 02 = 2)
Dst minute	minute in which the DLS time starts (w.r.t. the picture in Time field it is 00 = 0)

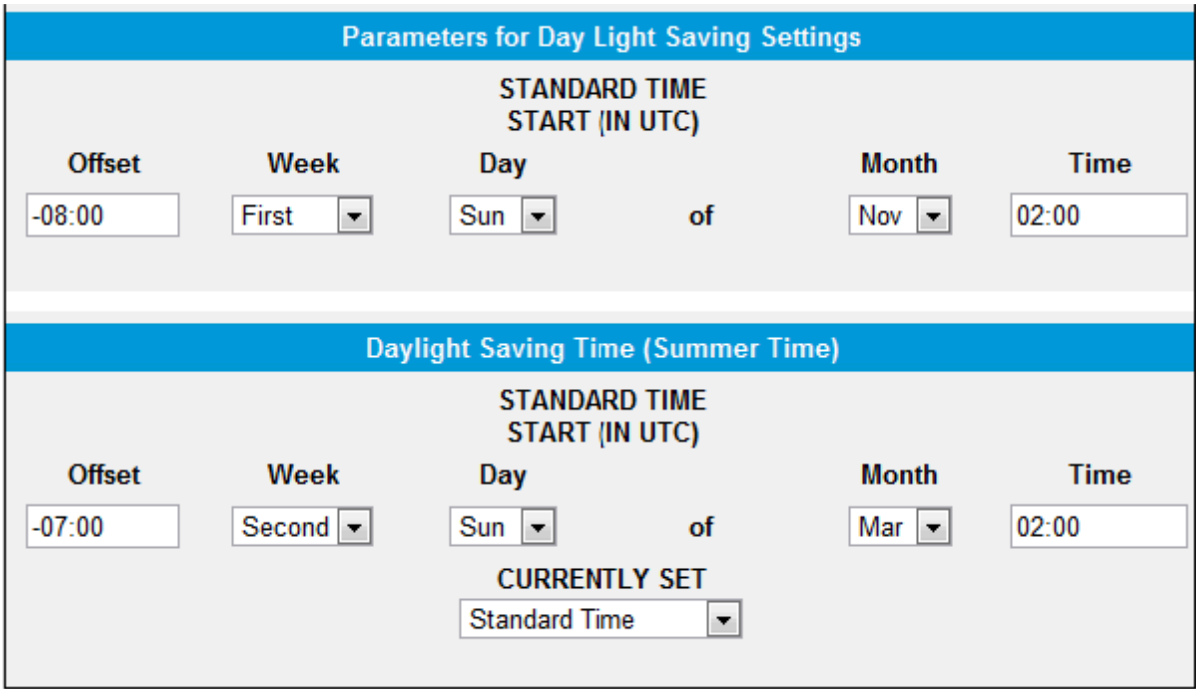


Figure 75

8.4 Device

Variables can be used to adjust specific device settings and obtain operational information.

Available System Memory	Returns the free available RAM memory in bytes; it is a 64 bit data; it is a read only variable.
Backlight Time	Returns the activation time in hours of the display backlight lamp since production of the unit; it is a read only variable.
Battery LED	Enable/Disable the use of the front LED indicator to report the low battery status. It can have values 0 (disabled) or 1 (enabled).
Battery Timeout	Reserved
Display Brightness	This variable is an integer of R/W type. Its range goes from 0 to 255. It can be used to check brightness level and adjust it from the application. Typical use is connected to a slider widget. When set to a low level (0..3), the backlight assume a low but visible value for around 8 seconds (to let the user change it otherwise nothing is visible on the display) and after that display appears as switched-off. However, even with a value of 0, the backlight is still on and the counter of backlight life time continues to increase.
External Timeout	Allows setting the non-operational time after which the display backlight is automatically turned off. The backlight is automatically turned back on when the user presses on the touchscreen. The variable is an int of R/W type. -1 = switch off backlight and disable touch (switch display off). Backlight counter is stopped. 0 = switch backlight on (so switch display on) 1..n = set a timeout for switch off backlight, so work like a screensaver timer
Flash Free Space	Returns the free space left in the device internal flash.
System Font List	List of system fonts. The variable is a read only string.
System Mode	Returns a value informing the operation status of the runtime. Possible values are: 1. Booting 2. Configuration mode 3. Operating mode 4. Restart 5. Shutdown
System UpTime	Returns the total time in hours in which the system has been powered since production of the unit. It is a read only variable.
Touch Buzzer, Buzzer Setup, Buzzer Control, Buzzer Off Time, Buzzer On Time	**Touch Buzzer:** allows enable/disable the touch audible feedback. It can have values 0 (disabled) or 1 (enabled). Starting from BSP 1.66.6 ARM / 2.73.1 MIPS, buzzer control has been extended as below: **Buzzer Setup** (replace Touch Buzzer System Variable) 0: disabled, no buzzer sound under any condition. 1: enabled, buzzer sounds as audible on any touchscreen event. 2: buzzer status controlled by System Variable "Buzzer Control" **Buzzer Control** 0: Buzzer off 1: Buzzer on 2: Buzzer blink (on and off times programmed by System Variables

Buzzer On and Off)

Buzzer Off Time
duration in milliseconds of off time when blink has been selected
minimum value: 100
maximum value: 5000
default value: 1000

Buzzer On Time
duration in milliseconds of on time when blink has been selected
minimum value: 100
maximum value: 5000
default value: 1000

8.5 Dump Information

Variables return information about the status of the copy process to external drives (USB or SD Card) for trend and archive buffers

Dump Trend Status	Returns value 1 during the copy process of the trend buffers. If the copy duration time is less than one second, the system variable does not change its value.
Dump Archive Status	Returns value 1 during the copy process of the archive buffers. If the copy duration time is less than one second, the system variable does not change its value.

8.6 Keypad

Variables return information about the status of the keypads.

Is keypad open	Returns value 1 if a keypad is open, 0 if no keypads are open.

8.7 Network

Variables allow you to show network device parameters. The network system variables are all strings in Read Only.

Gateway	Gateway address of the main Ethernet interface of device
IP Address	IP address of the main Ethernet interface of device
Mac ID	MAC ID of the main Ethernet interface of device
Subnet Mask	Subnet Mask of the main Ethernet interface of device

8.8 PLC

Variables return informations about the status of CODESYS 2.3 integrated in HMI panels.

PLC Status	Status of Integrated Codesys 2.3. 0 = RUN 1 = STOP 2 = RELOAD_FROM_FILE 3 = RESET_WARM 4 = RESET_COLD 5 = RESET_HARD This is a read only system variable.
Get CopyCodesysProject Action Status	Status of CopyCodesysProject Action related to intergated Codesys 2.3. 0 = ACTION_NOT_CALLED 1 = ACTION_IN_PROGRESS 2 = ACTION_COMPLETED 3 = ACTION_ABORTED_CHK_FILE_MISSING 4 = ACTION_ABORTED_PRG_FILE_MISSING 5 = ACTION_ABORTED_SDB_FILE_MISSING 6 = ACTION_ABORTED_MUTIPLE_CHK_FILES_FOUND 7 = ACTION_ABORTED_MUTIPLE_PRG_FILES_FOUND 8 = ACTION_ABORTED_MUTIPLE_SDB_FILES_FOUND 9 = ACTION_ABORTED_INCONSISTENT_FILE_NAMES 10 = ACTION_ABORTED_UNABLE_TOMAKE_TARGET_DIR 11 = ACTION_ABORTED_COPY_FAILED 12 = ACTION_ABORTED_CODESYS_MODULE_NOT_PRESENT This is a read only system variable.

8.9 Printing

Variables return information about the printing functions. All the variables are read only.

In the table below you can read the description of the available system variables.

Completion percentage	The percentage of competition of the current print job. It ranges from 0 to 100.
Current disk usage	The size (in bytes) of folder where PDF reports are stored (it is `reportspool` if option *Spool media type* is *Flash*).
Current job	The name of the report the job is processing. Current job is the following: - [report name] for a **Graphic Report** - [first line of text] for a **Text Report**
Current RAM usage	The size (in bytes) of the RAM used to process the current job.
Disk quota	The maximum size (in bytes) of the folder where PDF reports are stored.
Graphic job queue size	The number of the available graphic jobs in the printing queue.
RAM quota	The maximum size (in bytes) of the RAM used to generate reports.

Status	A string representing the status of the printing system. The possible values are **idle**, **error**, **paused** and **printing**.
Text job queue size	The number of the available text jobs in the printing queue.

8.10 Screen

Variables return information on the screen status.

Time remaining to unlock	Return time remaining to unlock screen (ref. LockScreen action)
X Screen resolution	It shows the x screen resolution of the display
Y Screen resolution	It shows the y screen resolution of the display

8.11 SD Card

Variables return information on the external SD Card plugged into the panel. They are 64 bit variables, except the drive name which is a string. All the variables are read only.

SD Card FreeSpace	Size in bytes of the available space.
SD Card Name	Name of the SD card.
SD Card Size	Size in bytes of the card plugged in the slot.
SD Card Status	Status of the SD card.

8.12 Server

Variables return information on the Server status. All following system variables refer to server, not to client (ex. JMobile Client).

Current page	Return the name of current page.
Current project	Return the name of current project.
Operating mode time	It shows the number of seconds elapsed after the last display start in operating mode.
Project load time	It shows the date time string in milliseconds, like the System Date format, when the project was loaded in runtime

8.13 Time

Variables return information on the System Time expressed in UTC format.
They are all Int (32 bits) of read/write type, except for the System time which is a 64 bit variable, still of read/write type. This is actually the UTC time which also is available as date/time from the other variables.

Day Of Month	Day of the month (1..31)
Day of Week	Day of the week (0=Sunday, .. , 6=Saturday)
Hour	Hour (0..23)
Minute	Minute (0..59)
Month	Current month (1..12)
Second	Second (0..59)
System Time	System time
Year	Current Year

8.14 USB Drive

Variables return information on the external USB drive connected to the panel; they are 64 bit variables, except the drive name which is a string. All the variables are read only.

USB Drive free space	Size in bytes of the available space
USB Drive Name	Name of the USB device
USB Drive Size	Size in bytes of the **device** plugged in the **USB port**
USB Drive Status	Status of the USB device

8.15 User Management

Variables return information on users and groups.

No of Remote-Clients Alive	Number of JMobile Client connected to the server. This is a read only short (16 bit).
This Client Group-Name	Name of the group to which the current logged user belongs to. This is a read only string.

This Client ID	The variable is valid with reference to the JMobile Client scope. Local and remote clients connected to the same "server" (same runtime) get a unique ID returned by this variable. This is a read only short (16 bit).
This Client User-Name	Name of the user logged to the Client where the system variable is displayed This is a read only string.

8.16 Version

Variables return information on OS & Runtime version.

Main OS Version	Return the version of Main OS. Ex. UN30HSxx60M0166
Runtime Version	Return the version of runtime. Ex. 1.90 (0) – Build (682)

9 Actions

Actions are the function used to interact with the system; they are normally executed when events are triggered.

When considering events generated by buttons (pressed or released) not all the actions are available for both states. In case the selected action is not supported for the actual state, the software will report a warning message as shown in the following figure.

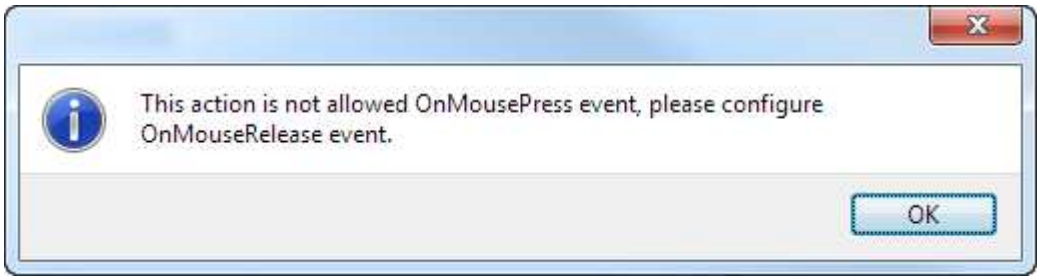


Figure 76

9.1 Widget Actions

The following chapters will include the description of a set of actions dedicated to handling widget visibility and control.

9.1.1 ShowWidget

The ShowWidget macro allows you to show or hide the page widgets. In the macro properties, select the widget you want to show or hide, then set the show properties as follows: false to hide and true to show widget.

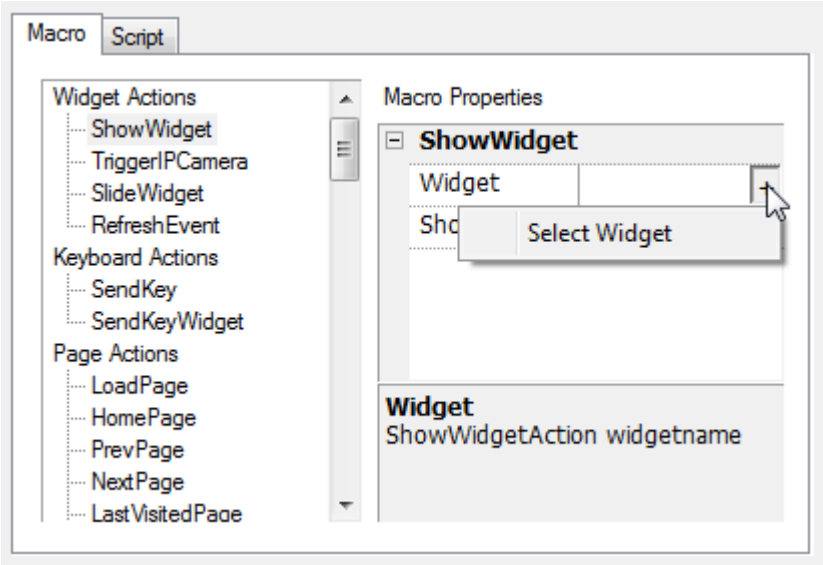


Figure 77

9.1.2 TriggerIPCamera

The TriggerIPCamera macro allows you to start the image capture from an IP Camera. Select the IP Camera Widget in the Macro Properties to trigger the capture from the IP Camera.

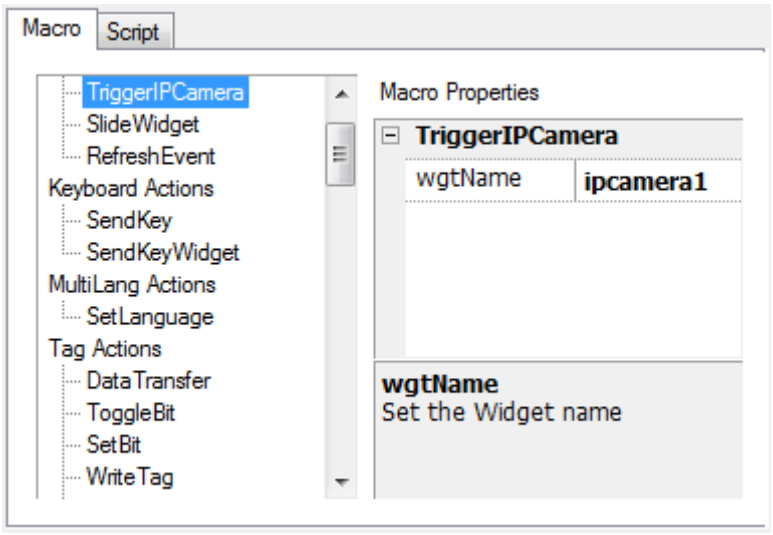


Figure 78

9.1.3 SlideWidget

The SlideWidget macro allows you to show the sliding effect of a Widget, or of a Widget group, in HMI Runtime.

NOTE *The widget or grouped widget can actually be outside of the page in the project and slide in and out of view.*

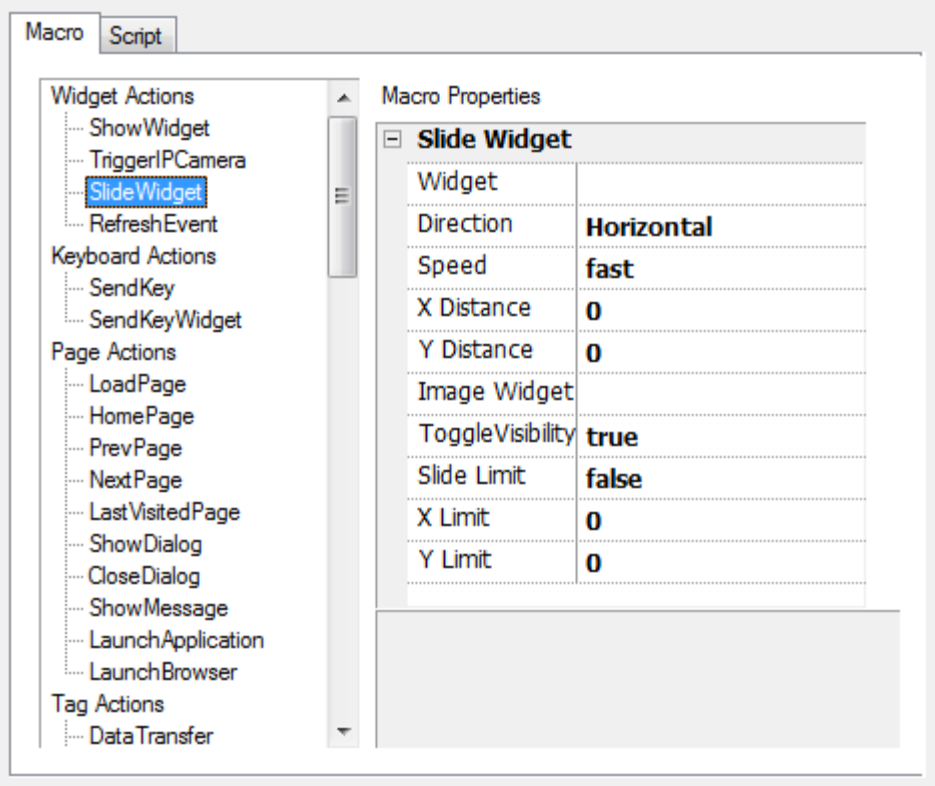


Figure 79

Widget
The Widget to slide

Direction
Sliding Direction

Speed
The transition speed of the sliding Widget

X Distance
The travel distance of the X coordinate of Pixel

Y Distance
The travel distance of the Y coordinate of Pixel

Slide Limit
Enable/Disable limiting the movement with Respect to the Coordinates (X and Y) of the Widget.

X Limit
When specified, automatically stops the slide action when the widget reaches the specified position.

Y Limit
When specified, automatically stops the slide action when the widget reaches the specified position.

Toggle Visibility
Toggle the Visibility of the Widget at the end of each Slide action.

Image Widget

Allows an image to show during the movement; the specified image will be shown during the Slide Operation between the start and end point of the movement.

9.1.4 RefreshEvent

The RefreshEvent macro allows you to refresh the selected Event Widget. The Event Widget is a component of the Alarm History Widget (see paragraph Alarms History Widget.).

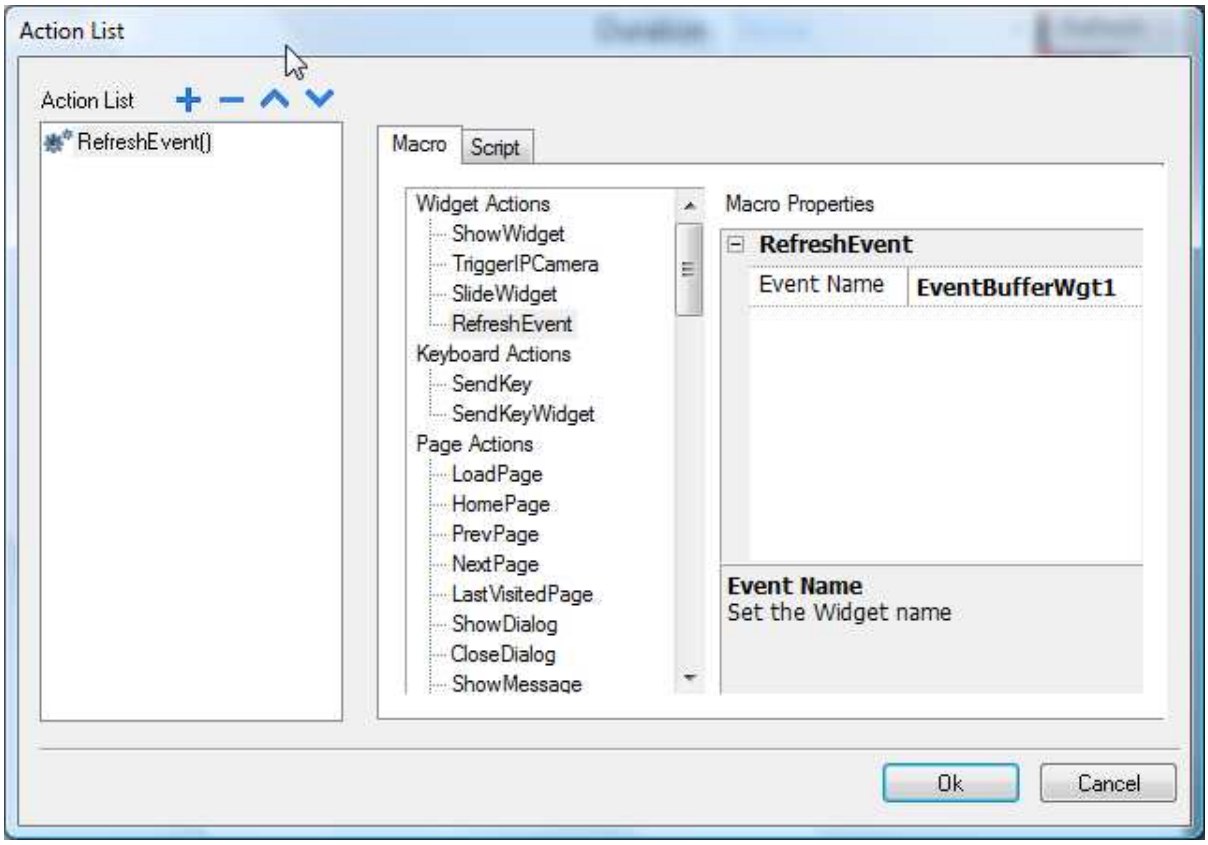


Figure 80

9.1.5 ContextMenu

Context menu is used to configure runtime parameters like Zoom level, to update runtime & project using an update package, to show the log window, to access the rotating menu for basic HMI configurations like IP Address or device local time etc. By default the context menu appears when the user press/click and hold for few seconds in the runtime area (in an area free of widgets like buttons). However, in the project area, it is possible to disable the OnHold event and configure the HMI to open the context menu just when the macro **ContextMenu** is called by the user. Usually this macro is attached to a button and protected to be used just by system administrators.

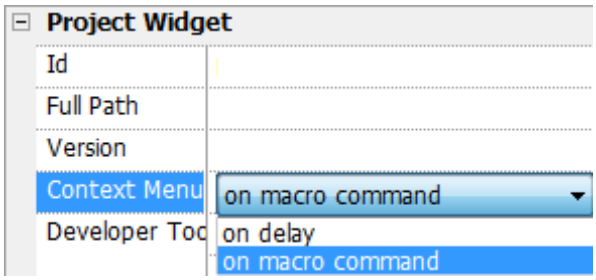


Figure 81

9.1.6 ReplaceMedia

ReplaceMedia macro is used at runtime to replace/update existing Media files of a project with new files provided via USB/SD or any other external device. Usually this macro is used to update project Images, Video or Music.

Following parameters are available:

- *MediaType*: Image, Video or Music. Define type of files to update.
- *Device:* define storage where runtime will search new media files (ex. \USBMemory).
- *sourcePath*: define folder where runtime will search new media files (ex. images).
- *Image Resize*: if enabled, the macro will resize images to the size of images at runtime. This is applicable for image media type only.
- *Silent*: if enabled, will execute replace media macro without user interaction. Otherwise a dialog will appear to allow user to select MediaType & storage where new media files are located (USB stick, SD Card etc).

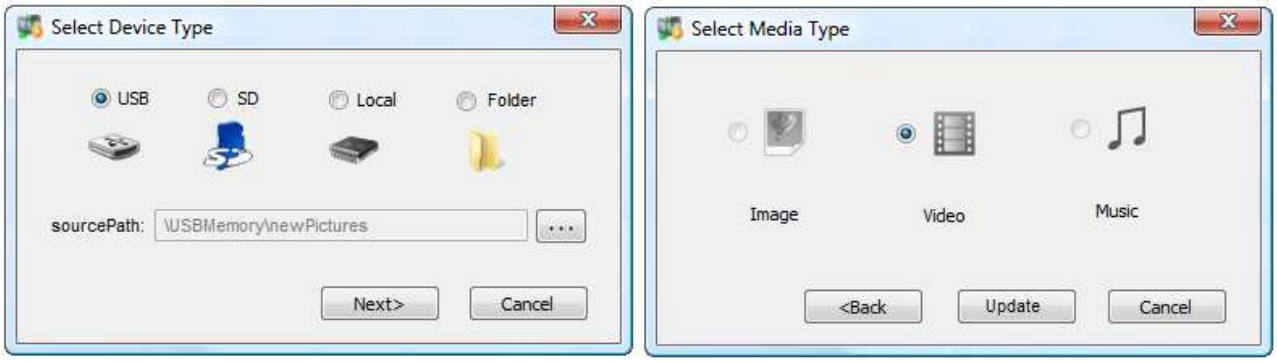


Figure 82

ReplaceMedia Macro is working in WCE and Win32 OS. When ReplaceMedia is executed in remote using JMobile Client or via ActiveX, files are searched on HMI local storages.

ReplaceMedia macro for Video & Music has been design to work in combination with MediaPlayer widget. ReplaceMedia macro for Images has been design to replace project images.

Follow API Javascript for replaceMedia method:

void replaceMedia(var sourcePath, var bSilent, var Device, var nMediaType, var bResize)
project.replaceMedia("Images", true, "\USBMemory", 1, true);

9.2 Media Actions

The Media Actions macros have been created to interact with the Media Player widget at Runtime.

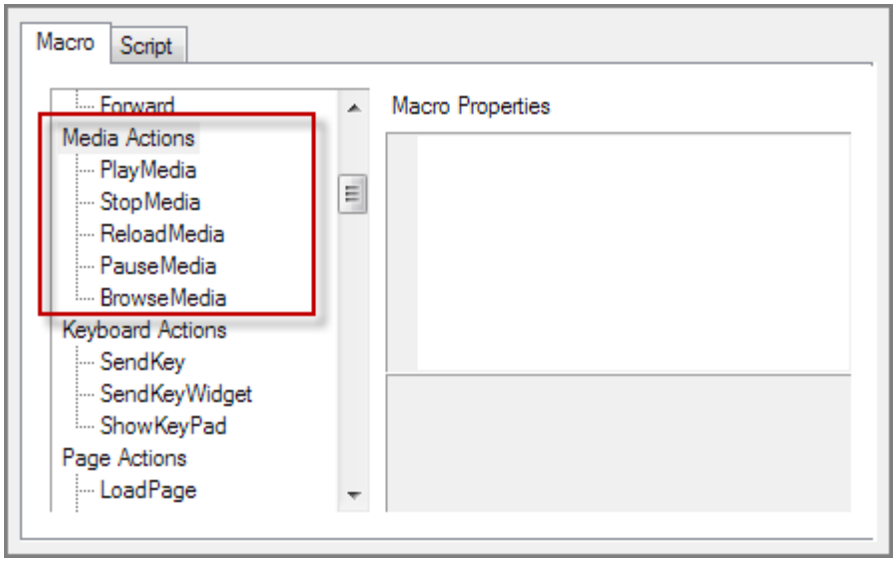


Figure 83

9.2.1 PlayMedia

The **PlayMedia** macro allows you to start playing the videos when the Media Player widget is paused or stopped. You have to specify the Media Player Widget in the macro properties.

9.2.2 StopMedia

The **StopMedia** macro allows you to stop the video running in the Media Player widget. You have to specify the Media Player Widget in the macro properties.

9.2.3 ReloadMedia

The **ReloadMedia** macro allows you to stop the video running in the Media Player widget and play it from the beginning. You have to specify the Media Player Widget in the macro properties.

9.2.4 PauseMedia

The **PauseMedia** macro allows you to pause the video running in the Media Player widget. You have to specify the Media Player Widget in the macro properties.

9.2.5 BrowseMedia

The **BrowseMedia** macro allows you to browse the HMI panel and select a video to be loaded in the Media Player widget. You have to specify the Media Player Widget in the macro properties.

9.3 Keyboard Actions

The Keyboard macro actions include Send Key and Send Key Widget.

9.3.1 SendKey

The SendKey macro is used to enter the predefined character to the Read/Write Widget. Define the predefined key code and Shift key code to the Macro actions property. In runtime, first click the R/W numeric Widget, then execute the Macro to send the predefined keys to the Numeric Widget. The action works on the field currently being edited.

NOTE *To use the SendKey macro, you must define the keypad type as **Macro** in the Numeric Widget properties*.

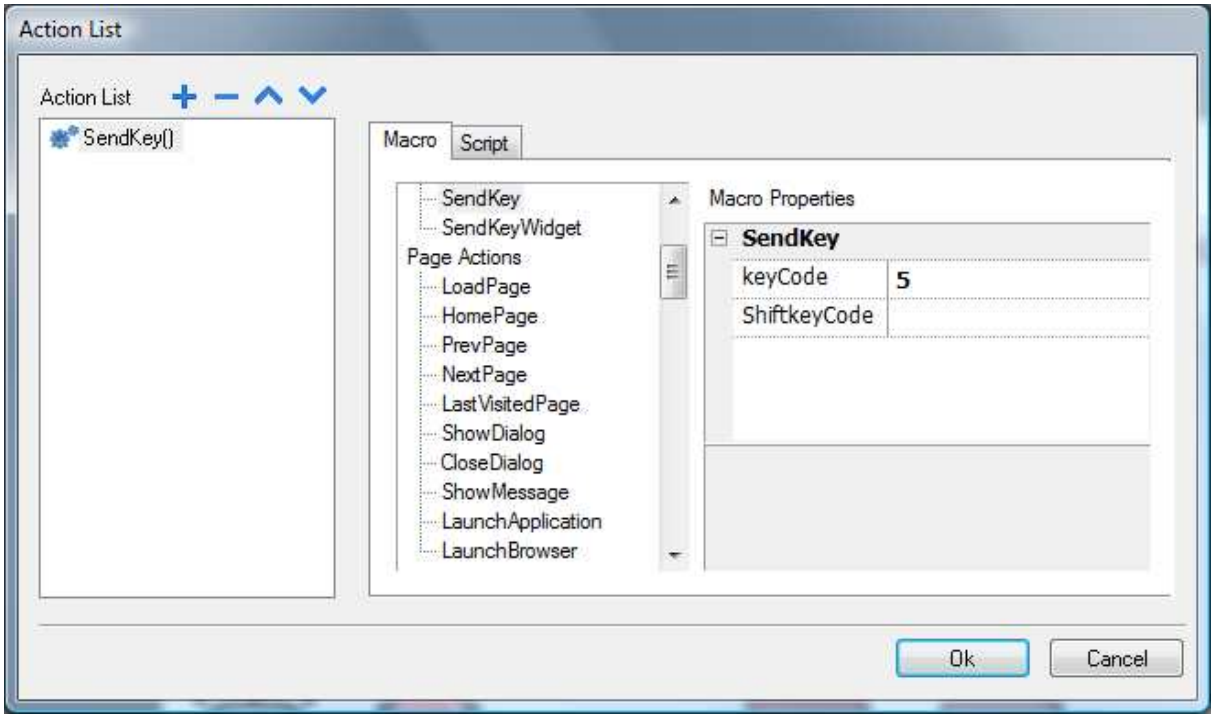


Figure 84

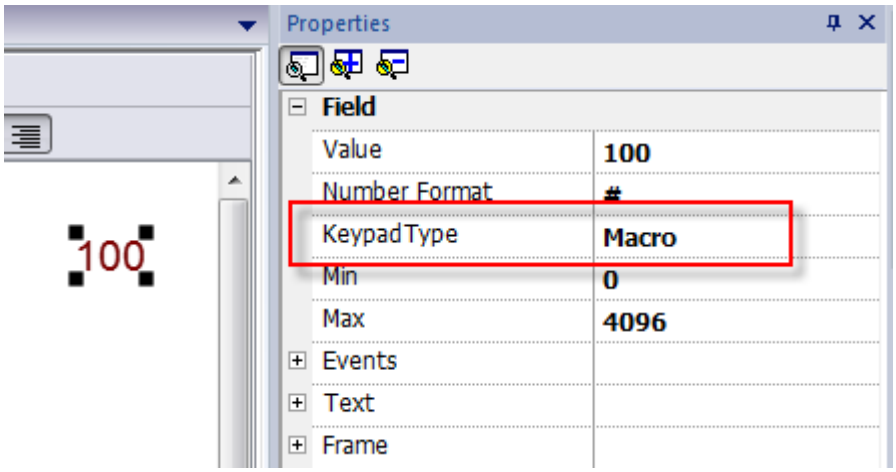


Figure 85

9.3.2 SendKeyWidget

The SendKeyWidget macro is used to enter the predefined character or function for a specific Widget. To use the macro, define the Widget ID and the key code in the Macro Properties.

The Control List Widget (available in the advanced category of the Widgets Gallery) is a good example of how this macro command can be used. Here Up and Down buttons have been implemented using the SendKeyWidget macro. See the figure below for reference.

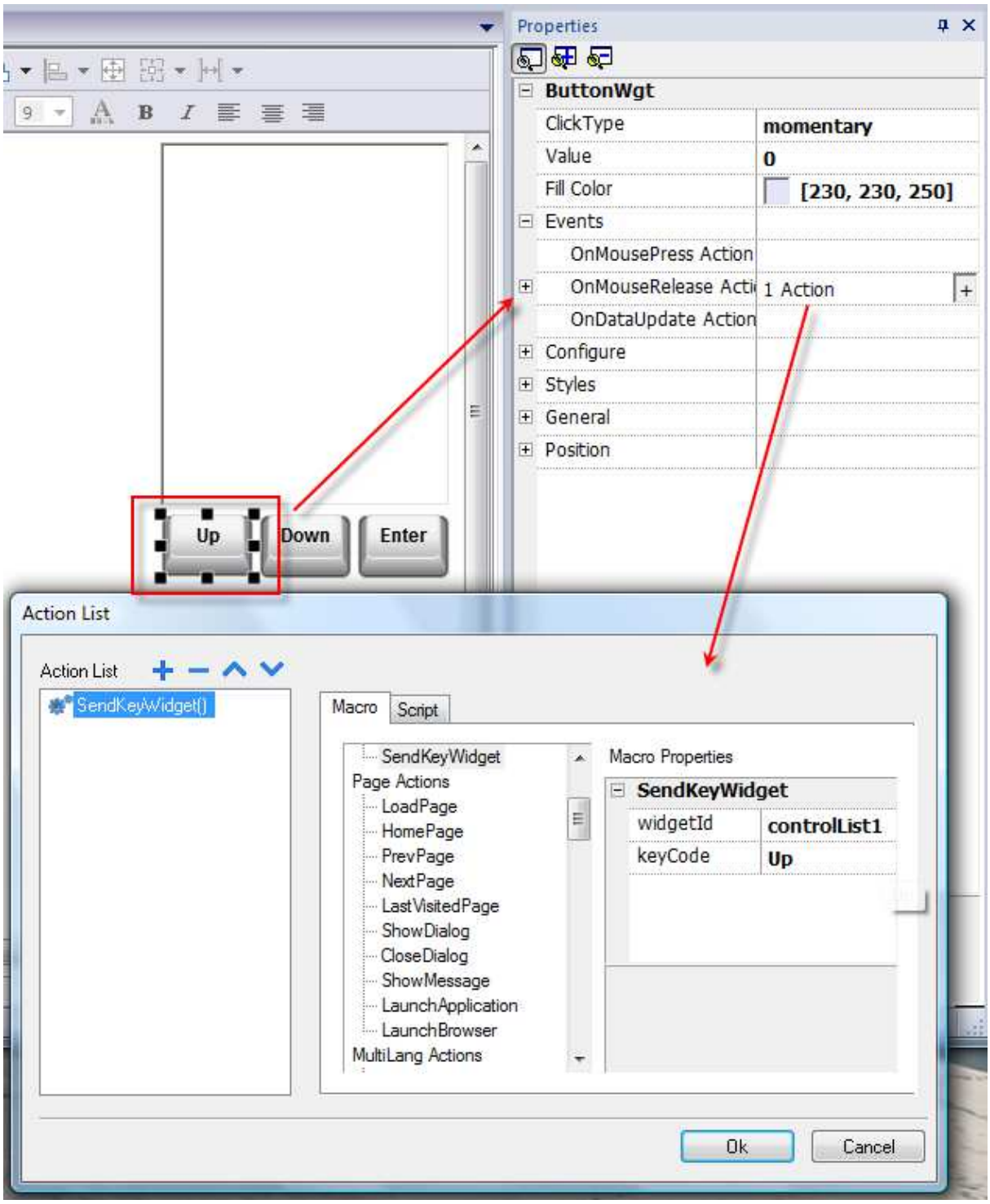


Figure 86

NOTE *To use the SendKey macro, you must define the keypad type as “Macro" in the Numeric Widget properties*.

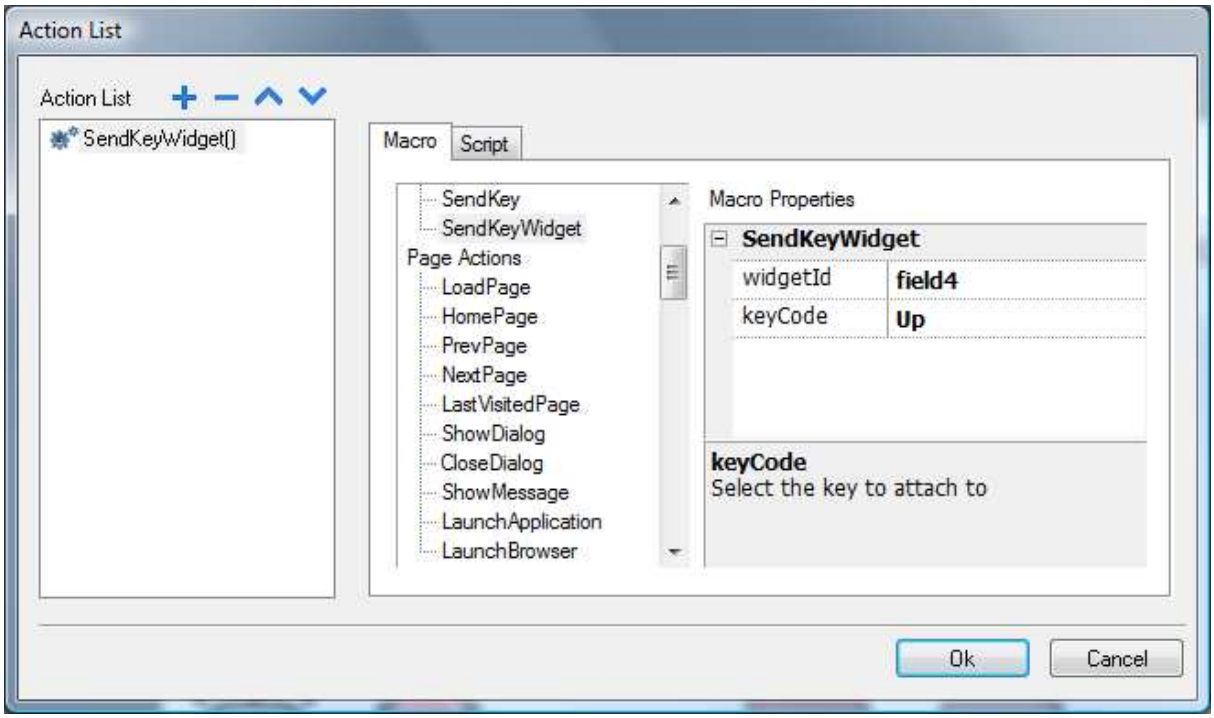


Figure 87

9.3.3 ShowKeyPad

ShowKeyPad is used to show the default operating system touch keypad. Some operating systems might not support it.

9.3.4 KeyboardMacros

KeyboardMacros enable and disable the use of keyboard Macros at runtime when using external keyboards.

You can also enable/disable macro execution related to keyboard studio side at the project level and at the level of the single page.

A dedicated property is available in the project property sheet and in the page property sheet.

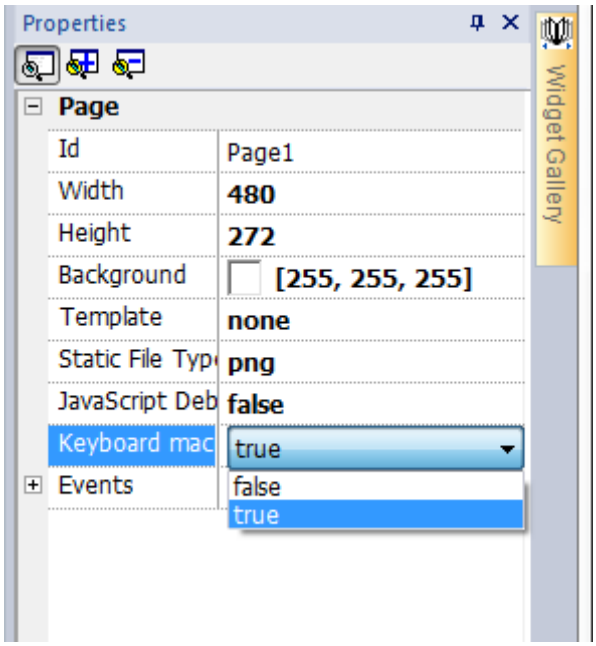


Figure 88

9.4 Page Actions

The Page Actions macros are used for page navigation and to load-specific pages. Please note that the Page Actions are programmable ONLY in the released state.
The Page Actions macros are available for Alarms, Schedulers and Mouse Release Events.

9.4.1 LoadPage

The LoadPage macro allows you to load the selected page of the project when the macro is executed.

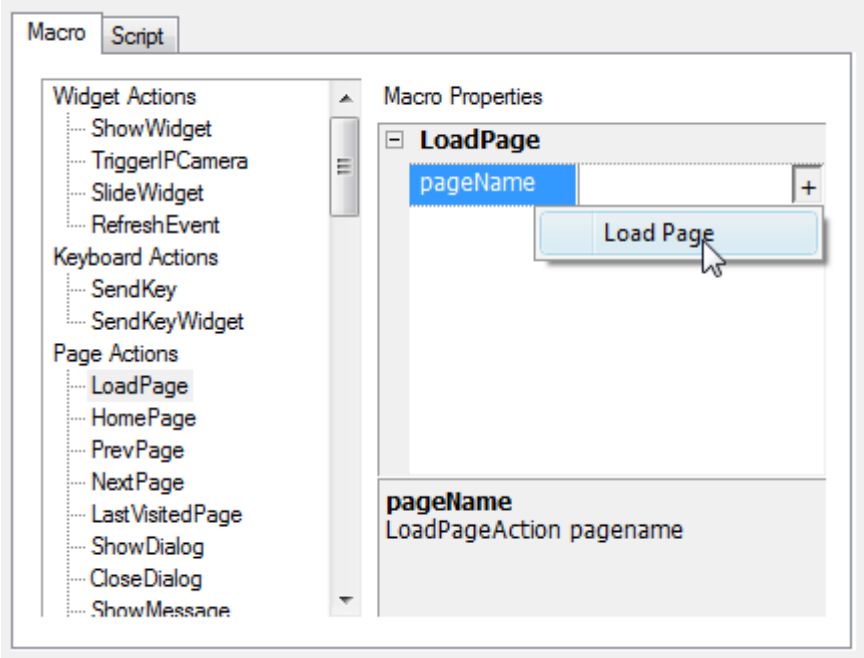


Figure 89

9.4.2 HomePage

The HomePage macro allows you to specify the home page.
By default, the home page is the first page of the project. However, you can change the home page in the project configuration properties. To change the home page, double click on the project name item in Project View. Once in Properties, choose the home page (as shown in the figure below).

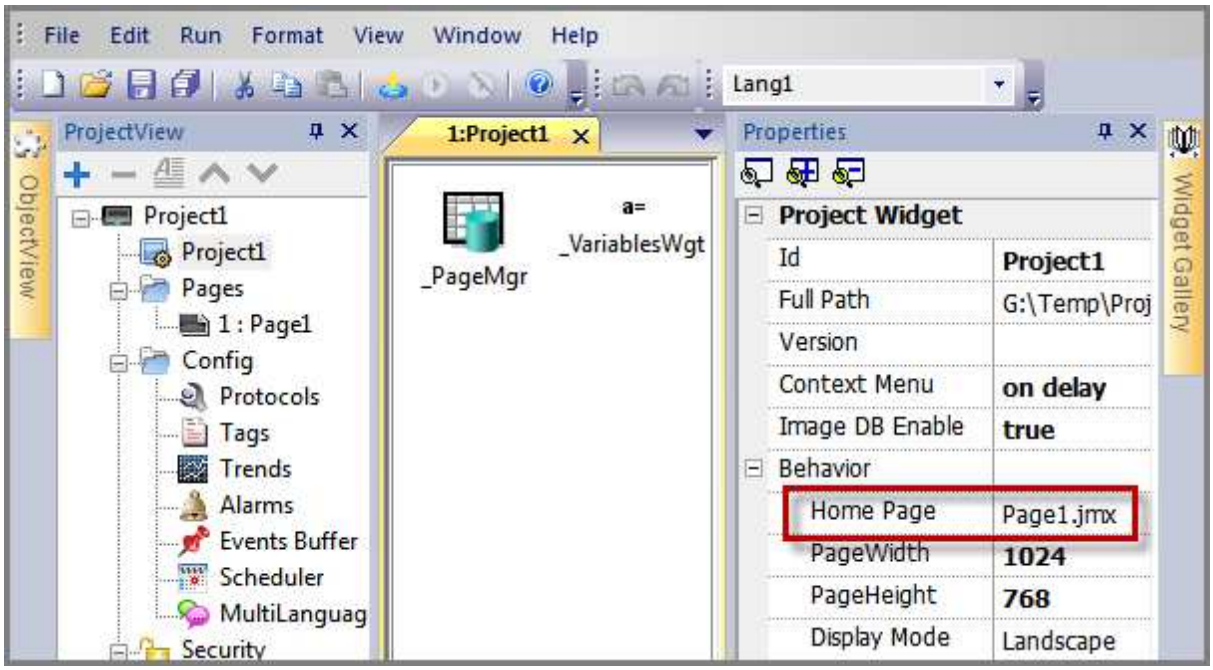


Figure 90

9.4.3 PrevPage

The PreviousPage macro allows you to navigate the HMI Runtime to the previous page.

9.4.4 NextPage

The NextPage macro allows you to navigate the HMI Runtime to the next page.

9.4.5 LastVisitedPage

The LastVisited page macro allows you to load the page previously displayed on HMI Runtime.

9.4.6 ShowDialog

The ShowDialog macro allows you to display the Dialog Pages defined in the project. After the execution of this macro, the HMI Runtime displays the specified Dialog Page.

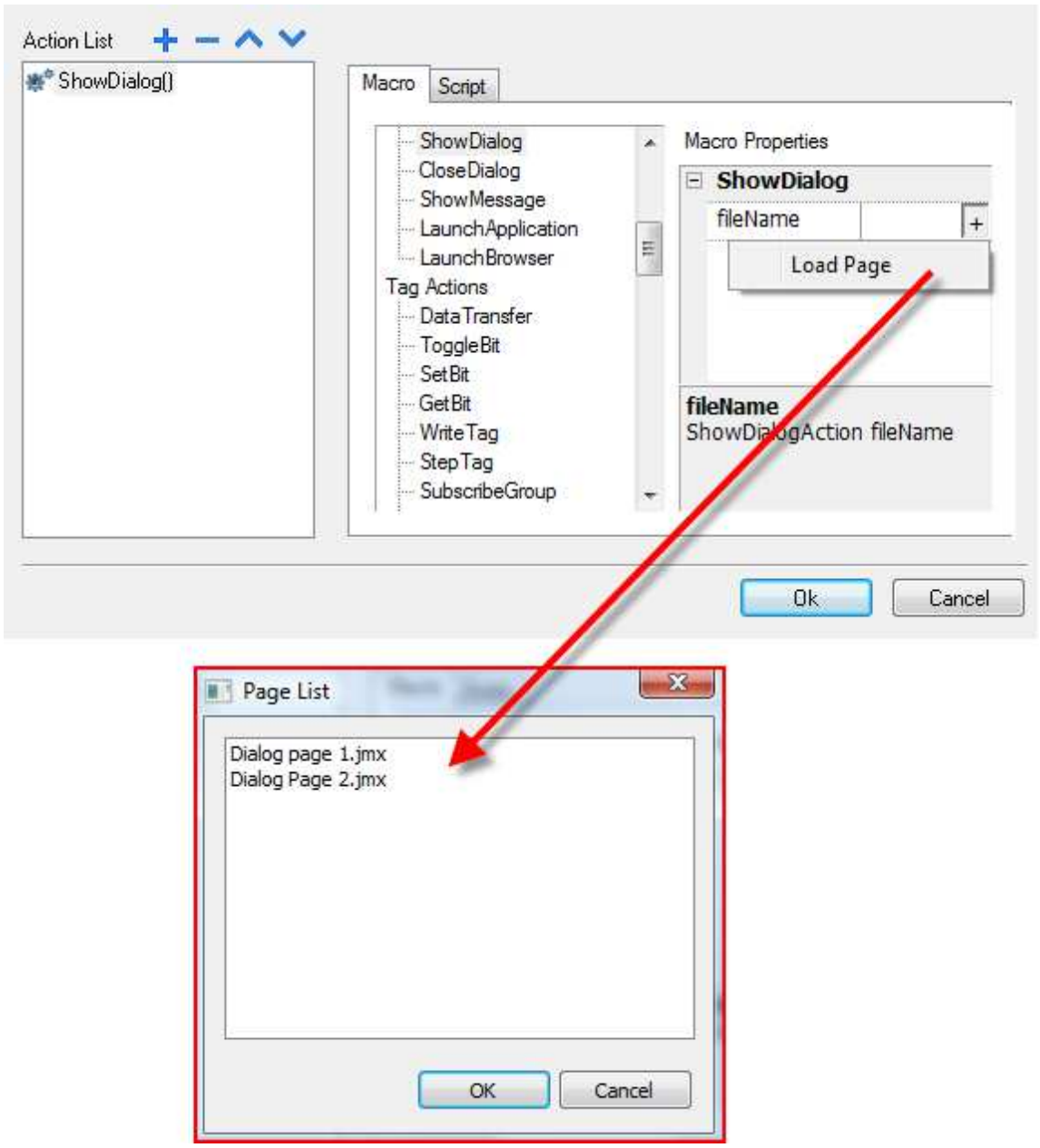


Figure 91

9.4.7 CloseDialog

The CloseDialog macro is applicable only to Dialog pages. The Close Dialog macro allows you to close the dialog page currently displayed.

9.4.8 ShowMessage

The ShowMessage macro allows you to display warning message popup when the macro is executed. Type the message that you wish to have displayed while executing the macro.

9.4.9 LaunchApplication

The LaunchApplication macro allows the user to launch an external application when the macro is executed. To configure, the following information must be provided in order to execute the requested application.

App Name
Name of the executable file complete with extension. For example, if you want to run Notepad application, the argument should be “notepad.exe"

Path
Application path; when the target platform is Windows CE, the path is \flash\qthmi. This is the folder that you see and have access to, when connecting to the panel via FTP.

Arguments
Some applications may need arguments to be passed. For example, to open a pdf file, specify the file name so that, while launching the application, the file name set in the argument is loaded on the application. For example, \flash\qthmi\Manual.pdf will open the document “Manual.pdf”.

Single Instance
This argument allows the application to start in a single instance or multiple instances. When single instance is selected, the system first verifies whether the application is already running. If it is running, then the application pulls it up (the operating system puts it in the foreground to the user’s attention); if it is not running, then the application is launched.

9.4.10 LaunchBrowser

LaunchBrowser opens the default web browser. You can define the URL address of the webpage in the arguments.

NOTE This macro only works in platforms that have a default web browser as application. *Not all platforms are equipped with a default web browser. For example, this macro is supported in OS based on Windows CE PRO with Internet Explorer enabled.*

9.4.11 LaunchVNC

Use this macro to execute VNC server configuration form. This macro is working only on embedded devices WCE based.

9.4.12 LaunchPDFViewer

Use this macro to run the PDF Viewer. This macro is working only in embedded devices based on Windows CE with ARM 1Ghz CPU.

Starting from v1.90 (b608), the PDF Reader has been added as software Plug-in (ref. to the chapter on software Plug-ins for more details) to allow developers of hmi applications to choose if enable & download it as part of the runtime. Just enable it from project properties -> plugins and install/update runtime to download it into the target.

9.4.13 LaunchUpdater

Use this macro to update project and/or runtime searching for it into an external device. Using “Path” parameter is possible to specify folder where searching the update package.

Examples of path could be:
- \USBMemory (for USB devices in WCE)
- \Storage Card (for SD devices in WCE)

NOTE *This macro is supported only in hmi panels based on WCE OS.*

9.4.14 LockScreen

Use this macro to lock touch screen temporary. Usually this action is used to allow users to cleanup touch screen without fire events pressing buttons or other widgets on page.

Screen → Time remaining to unlock system variable can be used to check time remaining to unlock screen.

9.5 MultiLang Actions

The Multi-Language (MultiLang) actions are used to select and modify the languages used in the application.

9.5.1 SetLanguage

The SetLanguage macro allows you to set the current display language. In Macro Properties, enter in the Language. At runtime, while executing the macro, the selected language will be applied to all applicable Widgets.

9.6 Tag Actions

The Tag Actions macros are used to interact with the application Tags.

9.6.1 DataTransfer

DataTransfer macros allow you to exchange data between two controllers, between registers within a controller, or from system variables to controllers (and vice versa). “SrcTag" refers to the source Tag and “DestTag" refers to Destination Tag. The various Tag types include a Controller Tag, a System Tag, a Recipe Tag and Widget Property.

9.6.2 ToggleBit

Toggle Bit macros allow you to “toggle" (meaning set or reset) a bit of a tag. The **BitIndex** allows you to select the bit to be inverted: this requires a read-modify-write operation; the read value is inverted and then written back to the controller tag.

9.6.3 SetBit

The SetBit macro allows you to set the selected bit. When the macro is executed, the value of the selected bit is set to “1".
The BitIndex property allows you to select the bit position inside the Tag.

9.6.4 ResetBit

The ResetBit macro allows you to reset the selected bit. When the macro is executed, the value of the selected bit value is set to “0”
The BitIndex property allows you to select the bit position inside the Tag.

9.6.5 WriteTag

WriteTag allows you to write constant values to the controller memory. In the Macro properties you have to specify the Tag name and the value to be written.

9.6.6 StepTag

The StepTag macro allows you to increment or decrement the value of a Tag.

TagName
Name of the Tag you want to Step

Step
Step value

Do not step over limit
Step Limit enables

Step Limit
If the “Do not step over limit” is true, then the macro will work until the Tag value reaches the specified value.

9.6.7 ActivateGroup

This macro activates tags update for a group of Tags.

Usually tags are updated when used in the current page (or always when defined in Tag Editor as Active =True). Using this macro it is possible to force the system to keep a group of tags always active (updated) independently if they are used on the current page or not.

9.6.8 DeactivateGroup

Deactivate a group of tags. Using this macro system stops reading a group of tags that had been previously activated

9.7 Trend Actions

Trend actions are used for both Live Data Trends and the Historical Trends Widget.

9.7.1 RefreshTrend

The **RefreshTrend** macro is used to refresh the Trend window. You have to specify the Trend Widget in the macro properties. This macro can be used in any of available graph widgets like Historical trends, Scatter Diagram and Consumption Meter.

9.7.2 Scroll Left Trend

The **ScrollLeftTrend** macro is used to scroll the Trend window to the left side, by one-tenth (1/10) of the page duration.

NOTE *With the Real-Time trend it is recommended pausing the trend using the macro **PauseTrend**, otherwise the window is continuously shifted to the current value.*

9.7.3 Scroll Right Trend

The **ScrollRightTrend** macro is used to scroll the Trend window to the right side, by one-tenth (1/10) of the page duration.

NOTE *With the Real-Time trend it is recommended pausing the trend using the macro **PauseTrend**, otherwise the window is continuously shifted to the current value.*

9.7.4 PageLeftTrend

The PageLeftTrend macro allows you to scroll the Trend window by one-page duration. For example, if the page duration is 10 minutes, then, with the PageLeftTrend macro you can scroll the trend left for 10 minutes.

9.7.5 Page Right Trend

The PageRightTrend allows you to scroll the Trend window by one-page duration. For example, if the page duration is 10 minutes, then, with the PageRightTrend macro, you can scroll the trend right for 10 minutes.

9.7.6 Page Duration Trend

The PageDuration macro is used to set the page duration of the Trend window. In Macro Properties, you must define the Trend Name and Page Duration.

NOTE You can also use a combo box Widget to select the page duration at Runtime.

9.7.7 Zoom In Trend

ZoomInTrend macro allows you to reduce the page duration.

9.7.8 ZoomOutTrend

ZoomOutTrend macro allows you to make the page duration longer.

9.7.9 Zoom Reset Trend

ZoomResetTrend macro allows you to reset the zoom level back to the original zoom level.

9.7.10 Pause Trend

PauseTrend macro allows you to stop plotting the Trend curves in the Trend window. When used with Real Time Trend the plotting stops when the curve reaches the right border of the graph. The Trend logging operation is not stopped from the panel when this macro command is used.

9.7.11 ResumeTrend

ResumeTrend macro allows you to resume a Trend plotting you previously paused. After executing the ResumeTrend macro, the Trend window will start to plot the data to the Trend once again.

9.7.12 Show Trend Cursor

The ShowTrendCursor macro allows the user to know the value of the curve at a given point on the X-Axis. Use this macro to activate the Trend Cursor. At Runtime, upon executing the macro, a Vertical Line (Cursor) will display in the Trend Widget. When the Graphic Cursor is enabled, the scrolling of the Trend is stopped. You can implement Scroll Cursor macros to move the Graphic Cursor over the curves, or to move the entire Trend window.

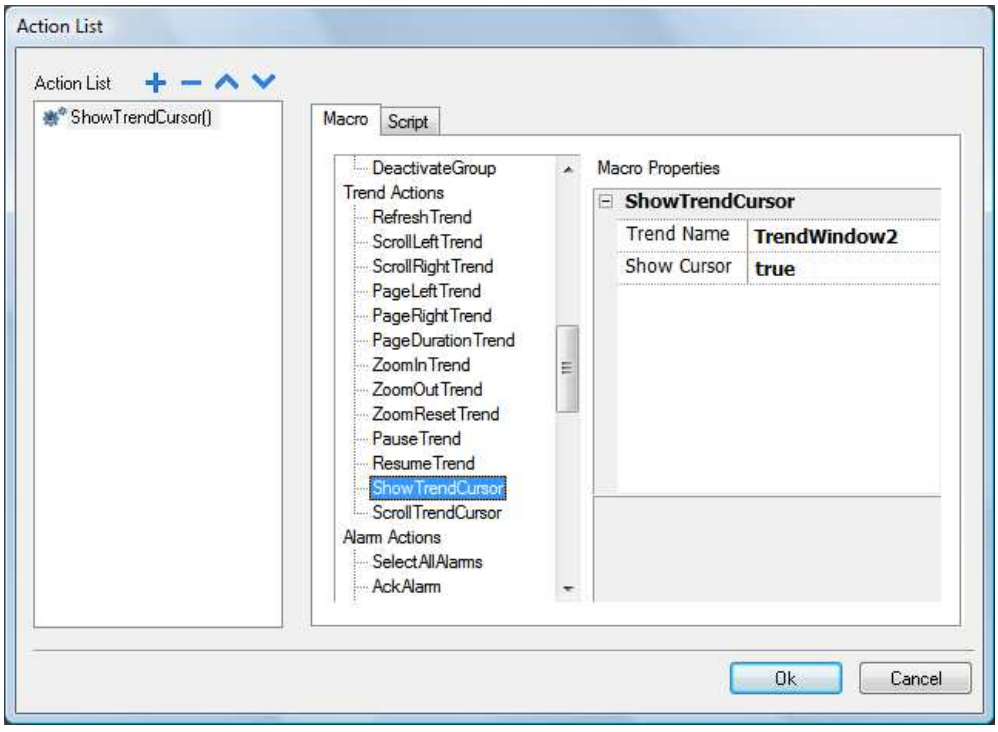


Figure 92

9.7.13 ScrollTrendCursor

The ScrollTrendCursor macro allows the user to scroll the Trend Cursor in forward or reverse time direction. The Y-Cursor value will display the Trend value at the point of the cursor. The scrolling percentage can be set at 1% or 10%. The percentage is calculated based on the Trend window duration.

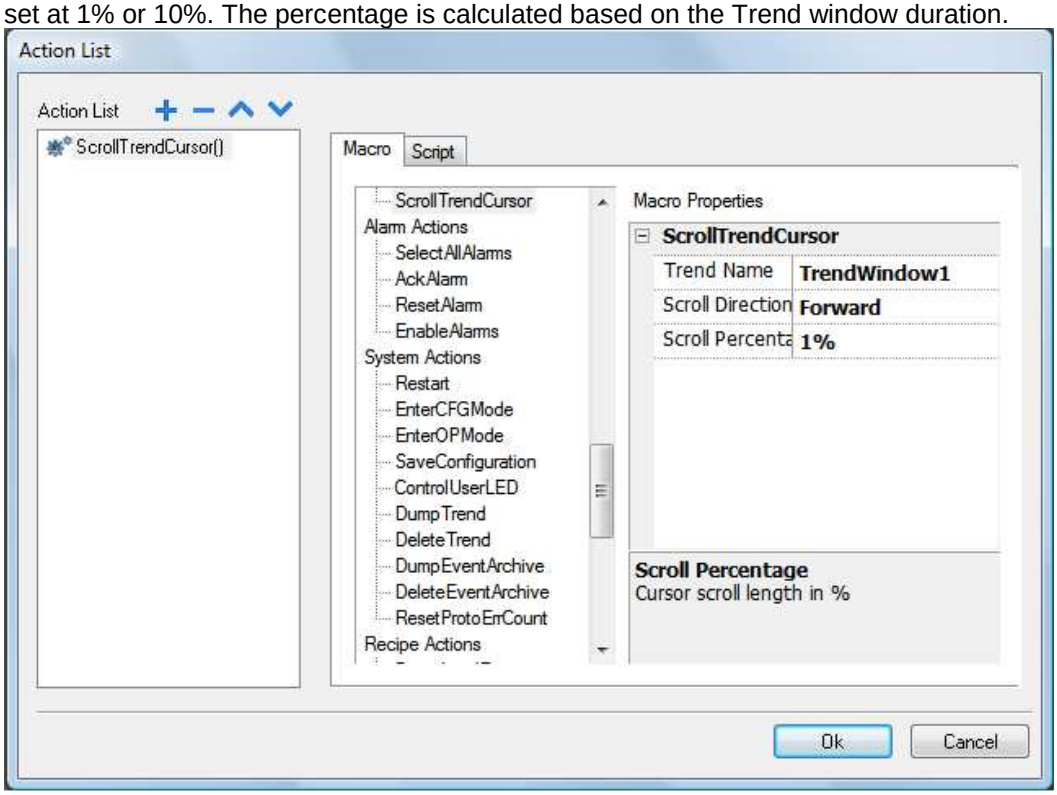


Figure 93

9.7.14 ScrollTrendtoTime

The **ScrollTrendtoTime** is used to scroll the Trend Window to a particular point in time. When you execute this macro the Trend Window will move to the time specified in the Macro Properties.

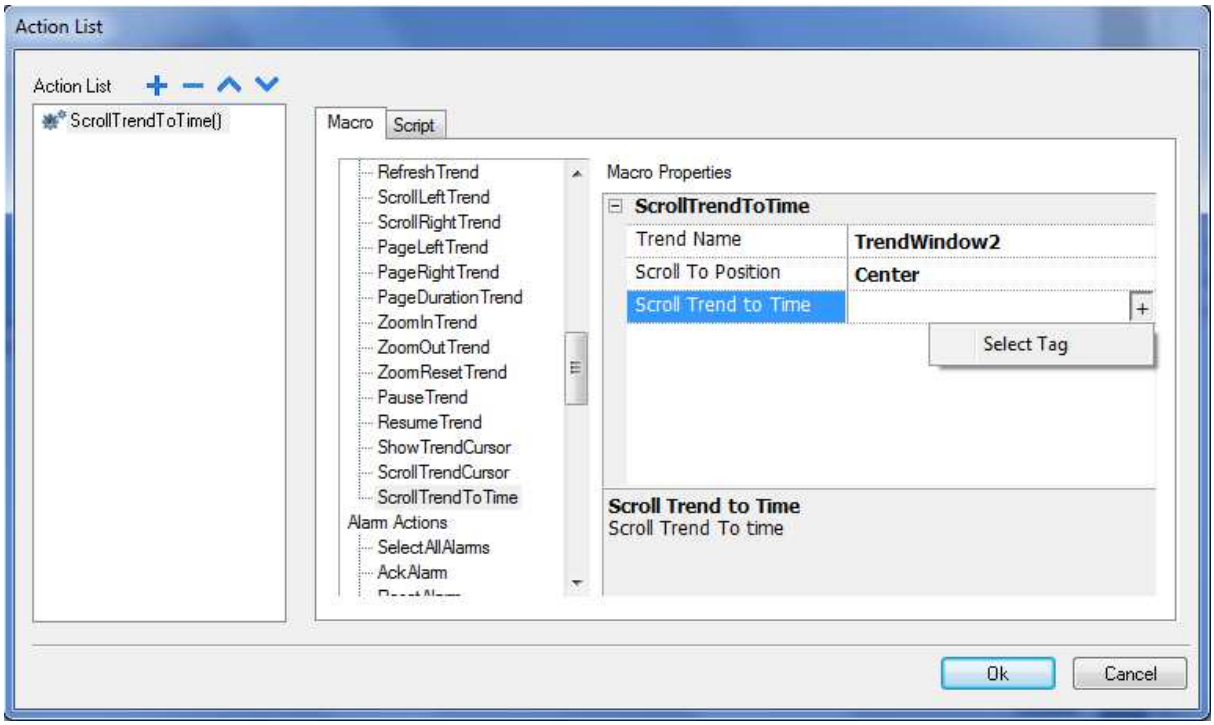


Figure 94

This Action may be very useful when you need to scroll at a specific position in a trend window based on the time at which a certain event occurs. This can be achieved by configuring an action for that alarm (event) that executes a Data Transfer of the system time into a Tag; when selecting that tag as “ScrollTrendtoTime” parameter (see above figure) the trend windows will be centered at the time in which the event has been triggered.

9.7.15 ConsumptionMeterPageScroll

The **ConsumptionMeterPageScroll** Macro is used to scroll page back/forward in ConsumptionMeter widget.

Available parameters are:

- **Trend Name:** Trend widget ID (ex. TrendWindow3)
- **Page Scroll Direction:** Forward/Reverse

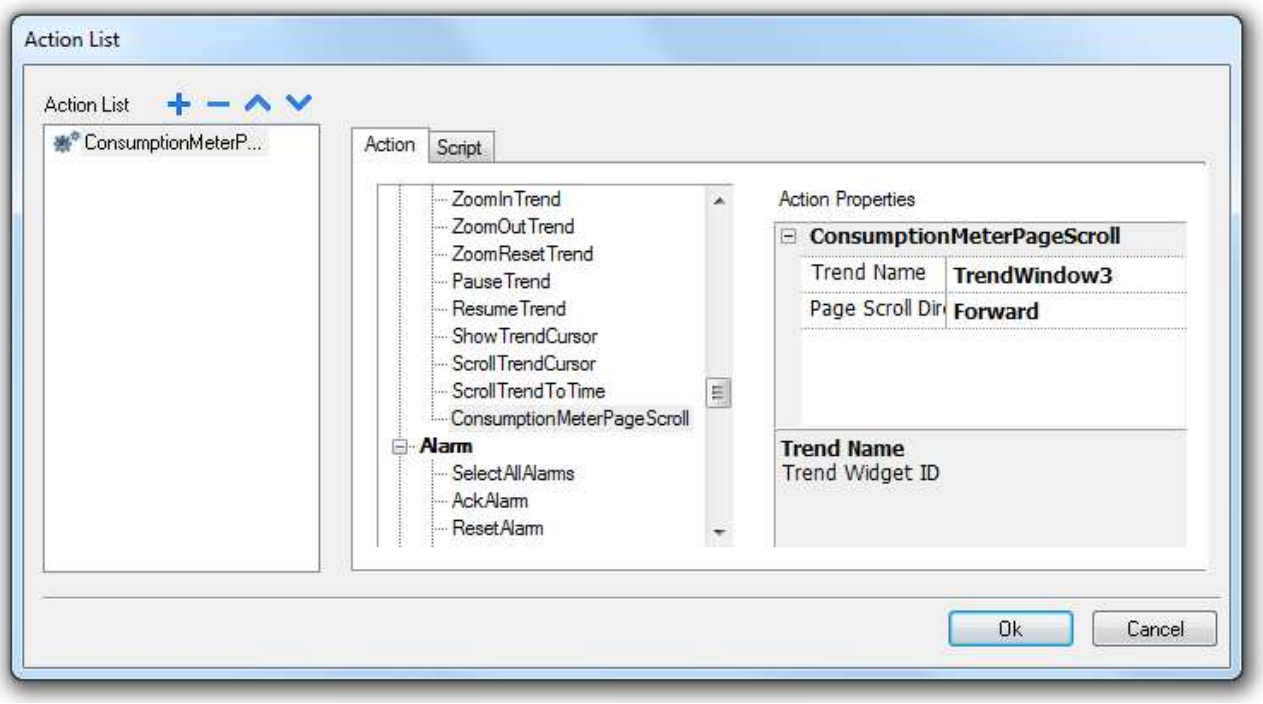


Figure 95

9.8 Alarm Actions

Alarm Actions are macros used to acknowledge or reset the alarms. The actions listed here can be used to build a custom Widget for the alarm display; you can observe an example of how these are used in the default Alarm Widget, available in the Widget gallery.

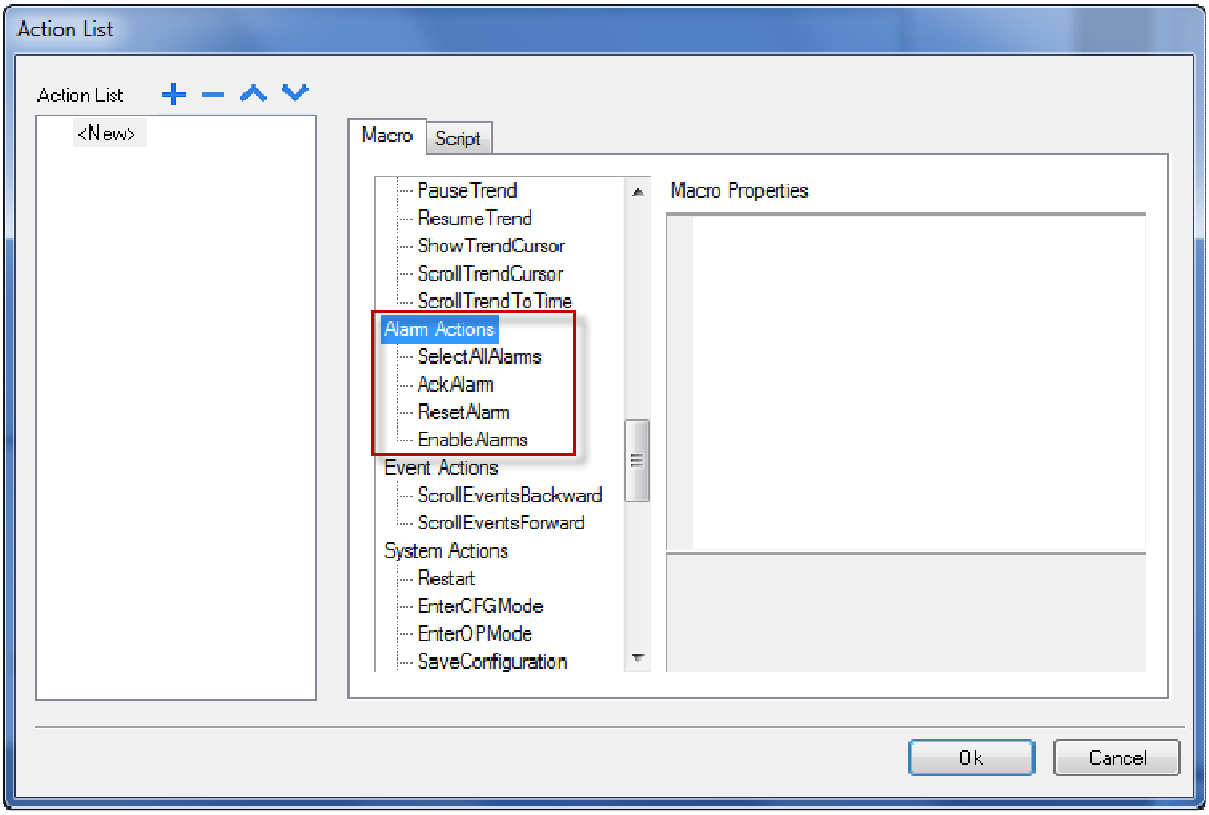


Figure 96

9.8.1 SelectAllAlarms

This macro allows you to select all the Alarms in the Alarm Widget.

9.8.2 AckAlarm

The AckAlarm macro allows for acknowledging the selected Alarms.

9.8.3 ResetAlarm

The ResetAlarm macro allows you to reset the selected acknowledged Alarms.

9.8.4 EnableAlarms

The EnableAlarms macro is used in conjunction with the “Save” button of the Alarm widget; it is required to properly save at runtime the changes made in the “Enable” check boxes from the “Enable” column in the alarm widget.

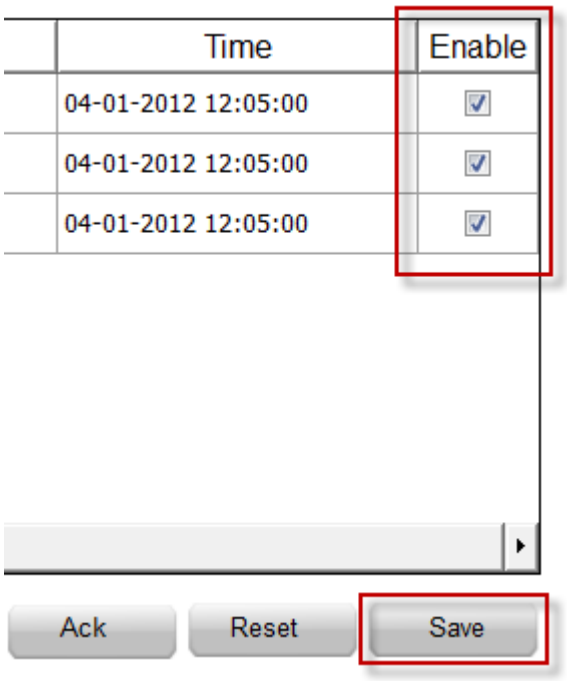


Figure 97

9.9 Event Actions

9.9.1 ScrollEventsBackward

Macro used by Alarm history widget to scroll events/alarms backward in table view (event buffer widget).

9.9.2 ScrollEventsForward

Macro used by Alarm history widget to scroll events/alarms forward in table view (event buffer widget).

9.10 System Actions

The System Actions macro allows you to use the system properties in Runtime.

9.10.1 Restart

The Restart system macro allows you to restart Runtime. After executing the macro, the Runtime goes to configuration mode and restart..

9.10.2 ControlUserLED

This allows the switching ON, OFF or blinking of the User LED. In the action properties, the LED action can be set to OFF, ON or Blink.

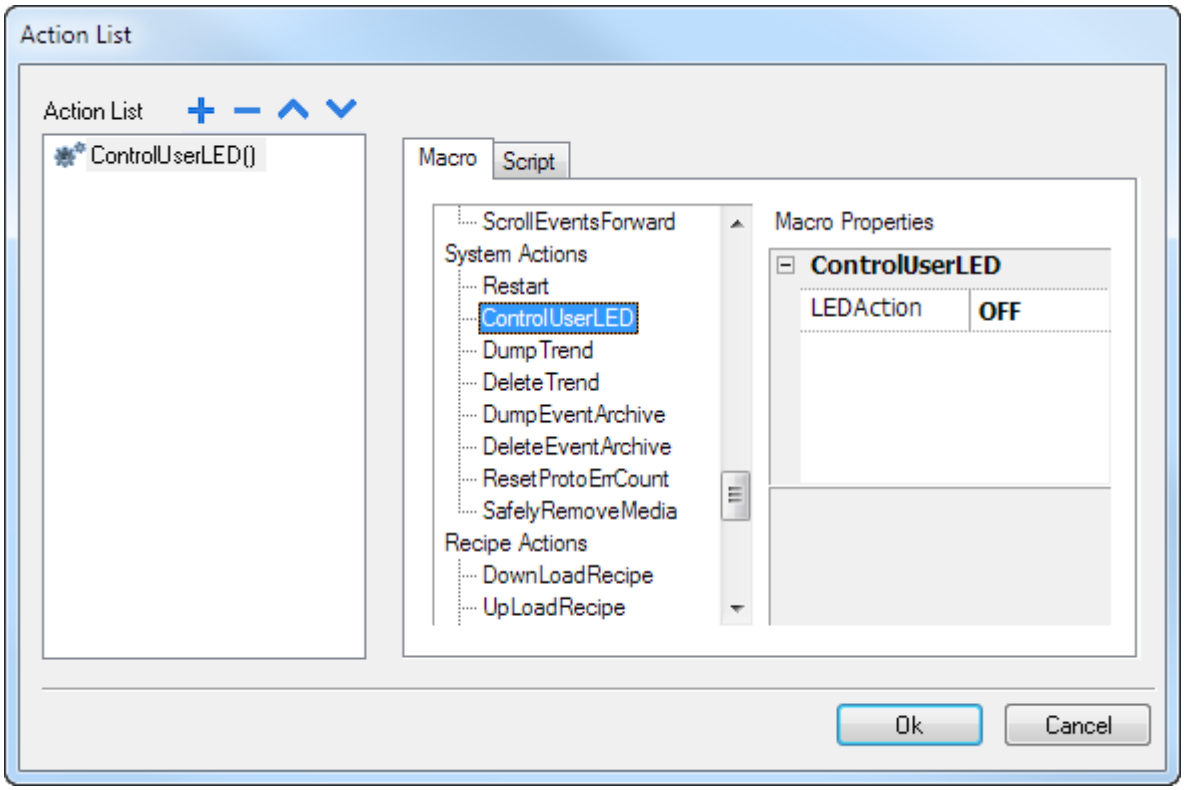


Figure 98

NOTE *Not all HMI models have User LED indicators. Check the product documentation for further information.*

9.10.3 DumpTrend

The **DumpTrend** macro is used to store the Historical Trend data to external drives, such as a USB drive or an SD card. In the macro properties, you must configure the Historical Trend name you want to store and the destination folder path. If you use a USB drive plugged into the USB port, the path will be `\USBMemory` or if you use an SD Card, the path will be `\Storage Card`, followed by the specified folder in the memory.

NOTE *The execution of the Dump action will automatically force a flush to disk of the data temporarily maintained in the RAM memory. See the chapter "Trend Editor" for further information about the policy used to save sampled data to disk.*

NOTE *The external drives plugged on the USB port of the panel must have format FAT or FAT32. NTFS format is not supported.*

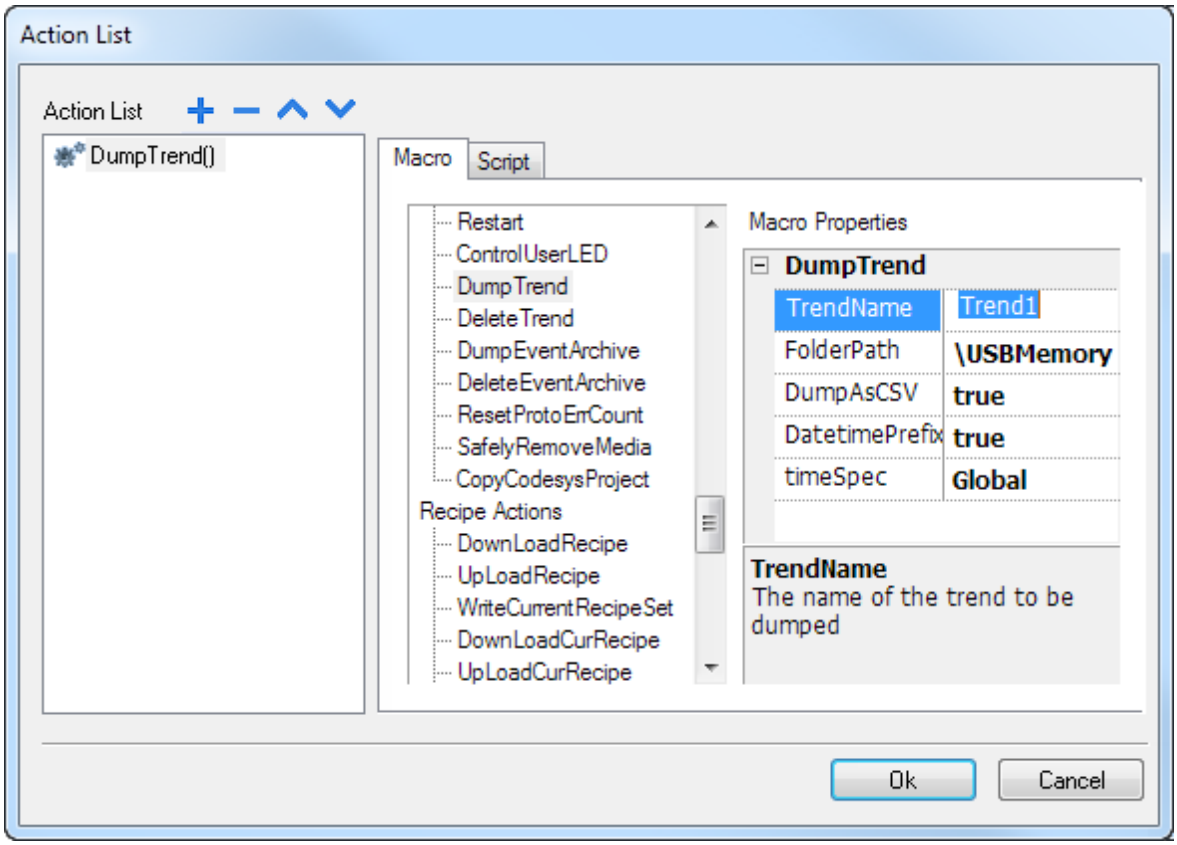


Figure 99

DumpAsCSV
If this option is set to true, then the buffer will be directly dumped to the specified location as a *.CSV file in the format specified below. If it is set as False, then the dump of the trend file will be in binary format; the result of the dump operation is actually a couple of files, one with extension .dat and one with extension .inf. An external utility is then required to convert it to a CSV format. These two files are both required by the utility to operate the conversion.

DateTimePrefixFileName
When this Option is enabled the dumped File will have the Date and Time as Prefix to the name of the File. For example if we are making a Dump at 10.10AM on 1-1-2012, then the file name will look like D2012_01_01_T10_10_Trend1.csv. [DYear_Month_day_THour_Minute_Filename]
This helps to know the Time at which the Dump was executed and also to identify which one is the latest.

timeSpec
This option defines the time format used when dumping the trend to file.

- **Local**: the time values exported are the time of the HMI device.
- **Global**: the time values exported are in the Coordinated Universal Time (UTC) format.

Example:

Local	2012-10-11T05:13:43.724-07:00
Global	2012-10-11T12:13:43.724Z

NOTE *The software tool required to convert the dump files to CSV are available in the JMobile Studio folder called "Utils" under the directory where the software is installed.*

The tool needed to convert trend buffers is called "TrendBufferReader.exe".

The TrendBufferReader.exe tool can be invoked using a batch file with the following syntax:

TrendBufferReader -r Trend1 Trend1.csv 1

where Trend1 is the name of the trend buffer without extension resulting from the dump (original file name is trend1.dat) and Trend1.csv is the name desired for the output file.

The resulting CSV file has 5 columns with the following meaning:

Data Type, **Value**, **Timestamp**(UTC), **Sampling Time**(ms), **Quality**

Where:

Data Type:
Code that gives information about the data type of the sampled Tag according to the following codes:

Code	Type
0	Empty
1	Boolean
2	Byte
3	Short
4	Int
5	Unsigned Byte
6	Unsigned Short
7	Unsigned Int
8	Float
9	Double

Value
Value of the sample

Timestamp(UTC)
Timestamp in UTC format

SamplingTime(ms)
Sampling interval time in milliseconds

Quality
Gives information on the tag value quality. The information is coded according the OPC DA standard; the information is stored in a byte data (8 bits) currently defined in the form of three bit fields; Quality, Sub status and Limit status. The 8 Quality bits are arranged as follows: QQSSSSLL

For a complete and detailed description of all the single fields, please refer to the OPC DA official documentation. Shown below are the most commonly used quality values returned by the HMI acquisition engine:

Quality Code	Quality	Description
0	BAD	The value is bad but no specific reason is known

4	BAD	There is some server specific problem with the configuration. For example, the tag in question has been deleted from the configuration file (tags.xml).
8	BAD	This quality may reflect that no value is available at this time, for reasons such as the value may have not been provided by the data source.
12	BAD	A device failure has been detected
16	BAD	Timeout occurred before device responded.
24	BAD	Communications have failed.
28	BAD	There are no data found to provide upper or lower bound value (trend interface specific flag).
32	BAD	No data have been collected (i.e. archiving not active. Trend interface specific flag). When the HMI return online after a reboot or from a condition where sampling stopped, a sample with quality value 32 is added to indicate this temporary offline status.
64	UNCERTAIN	There is no specific reason why the value is uncertain.
65	UNCERTAIN	There is no specific reason why the value is uncertain. (The value has ‘pegged’ at some lower limit)
66	UNCERTAIN	There is no specific reason why the value is uncertain. (The value has ‘pegged’ at some high limit.)
67	UNCERTAIN	There is no specific reason why the value is uncertain. (The value is a constant and cannot move.)
84	UNCERTAIN	The returned value is outside the limits defined for it. Note that in this case the “Limits” field indicates which limit has been exceeded but the value can move farther out of this range.
85	UNCERTAIN	The returned value is outside the limits defined for it. Note that in this case the “Limits” field indicates which limit has been exceeded but the value can move farther out of this range. (The value has ‘pegged’ at some lower limit)
86	UNCERTAIN	The returned value is outside the limits defined for it. Note that in this case the “Limits” field indicates which limit has been exceeded but the value can move farther out of this range. (The value has ‘pegged’ at some high limit.)
87	UNCERTAIN	The returned value is outside the limits defined for it. Note that in this case the “Limits” field indicates which limit has been exceeded but the value can move farther out of this range. (The value is a constant and cannot move.)
192	GOOD	

9.10.4 DeleteTrend

The DeleteTrend macro allows you to delete saved Trend data from the file. In Macro Properties, define the Trend name from which you want to delete the trend logs.

9.10.5 DumpEventArchive

The DumpEventArchive macro is used to export the Historical Alarm log and Audit Trail data to external drives, such as a USB memory or SD card. If you use a USB drive the path will be `\USBMemory` or if you use an SD Card the path will be `\Storage Card`, followed by the specified folder in the memory.

NOTE *The external drives plugged on the USB port of the panel must have format FAT or FAT32. NTFS format is not supported.*

In the Macro Properties, you need to configure the Event buffer name that you want to dump and the destination folder path. The **DumpConfigFile** property must be set to true when you plan to convert the dumped files to CSV.

DumpAsCSV
If this option is set to true, the buffer will be directly dumped to the specified location as a *.CSV file. If it is set as false, then the dump of the trend file will be in binary format an external tool is then required to convert it to a CSV format.

DateTimePrefixFileName
When this option is enabled the dumped file will have the Date and Time as prefix to the name of the file. For example if we are making a Dump at 10.10AM on 1-1-2012, then the file name will look like D2012_01_01_T10_10_alarmBuffer1.csv. [DYear_Month_day_THour_Minute_Filename]
This helps to know the Time at which the Dump was made and also to identify which one is the latest

NOTE *This option is only supported when exporting to CSV directly.*

timeSpec
This option defines the time format used when dumping the event archive to file.

- **Local**: the time values exported are the time of the HMI device.
- **Global**: the time values exported are in the Coordinated Universal Time (UTC) format.

Example:

- Local 2012-10-11T05:13:43.724-07:00
- Global 2012-10-11T12:13:43.724Z

When exporting Event buffers in binary format assuming the DumpConfigFile option is set to true (recommended settings), the result of the dump action execution is 2 folders; one is called “data” and it contains the data files, the second one is called “config” and it does contain the configuration files needed by the tool to recover the complete information for proper conversion to CSV.
Once the two folders are copied from the root of the USB drive to the computer disk, the folder structure will look as follows:

```
.\config\
        alarms.xml
        eventconfig.xml
.\data\
        AlarmBuffer1.dat
        AlarmBuffer1.inf
.\
AlarmBufferReader.exe
```

NOTE *The utility is distributed in JMobile Studio in the folder Exor\JMobile Suite\Utils.*

The AlarmBufferReader can be called from command line with the following syntax:

AlarmBufferReader AlarmBuffer1 FILE ./AlarmBuffer1.csv

Where AlarmBuffer1 is the name of the dumped .dat file without extension and AlarmBuffer1.csv is the desired output file name.

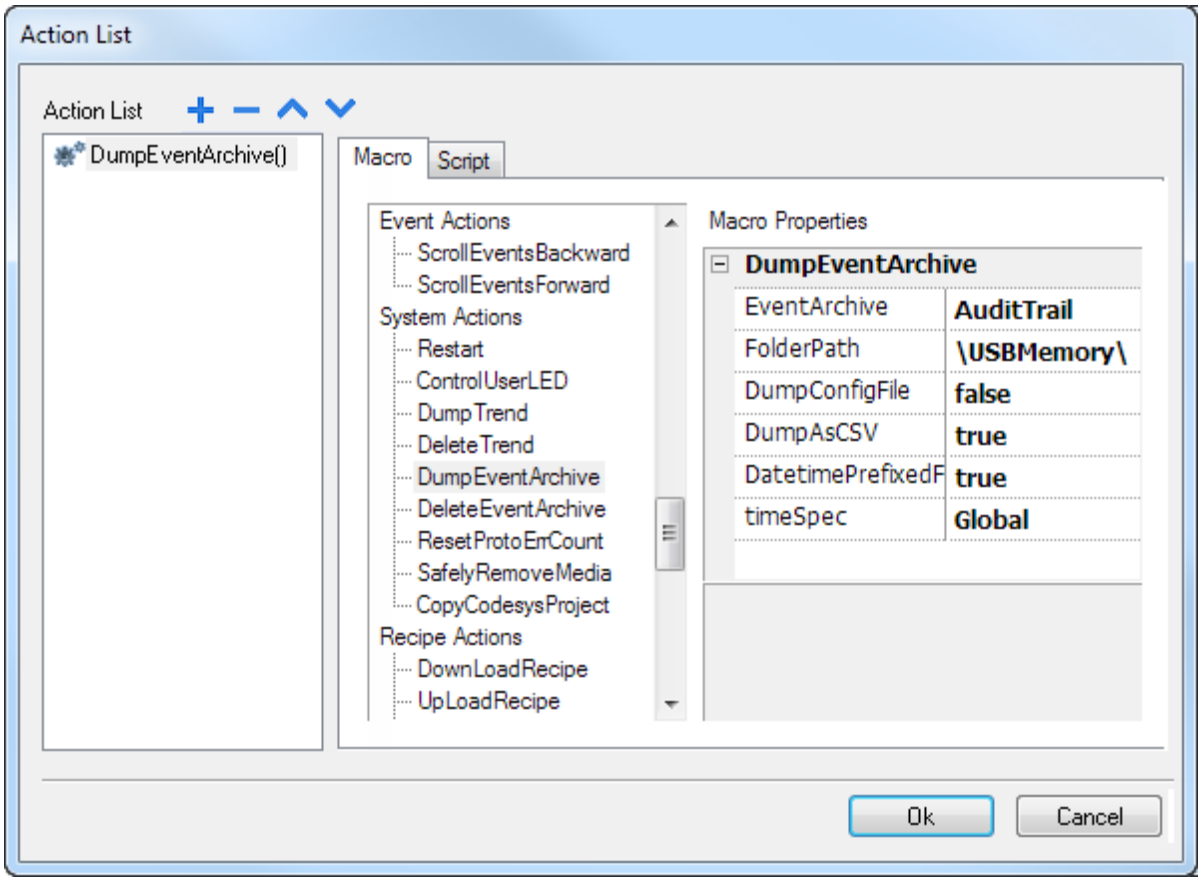


Figure 100

The utility called “AuditTrailBufferReader.exe" is available for Audit Trail buffers.

NOTE *The action must to be configured with the option DumpConfigFile set to true.*

The result of the dump is a directory structure similar to the one generated for Events.
The conversion tool can be called from the command line according to the following syntax:

AuditTrailBufferReader AuditTrail FILE ./AuditTrail.csv

Where AuditTrail is the name of the dumped buffer without extension and AuditTrail1.csv is the desired output file name.

9.10.6 DeleteEventArchive

The DeleteEventArchive macro allows you to delete saved Event buffers log data from the file. In the macro properties, define the Event buffer name that you want to delete from the Event logs.

9.10.7 ResetProtoErrCount

The ResetProtoErrCount macro is used to reset the Protocol Error Count System Variable. See the chapter “System Variables“for further information about system variables.

9.10.8 SafelyRemoveMedia

If you unplug an SD Card or a USB drive from the HMI while it is transferring or saving information, you risk losing some information. This macro provides a way to help you safely remove such devices.

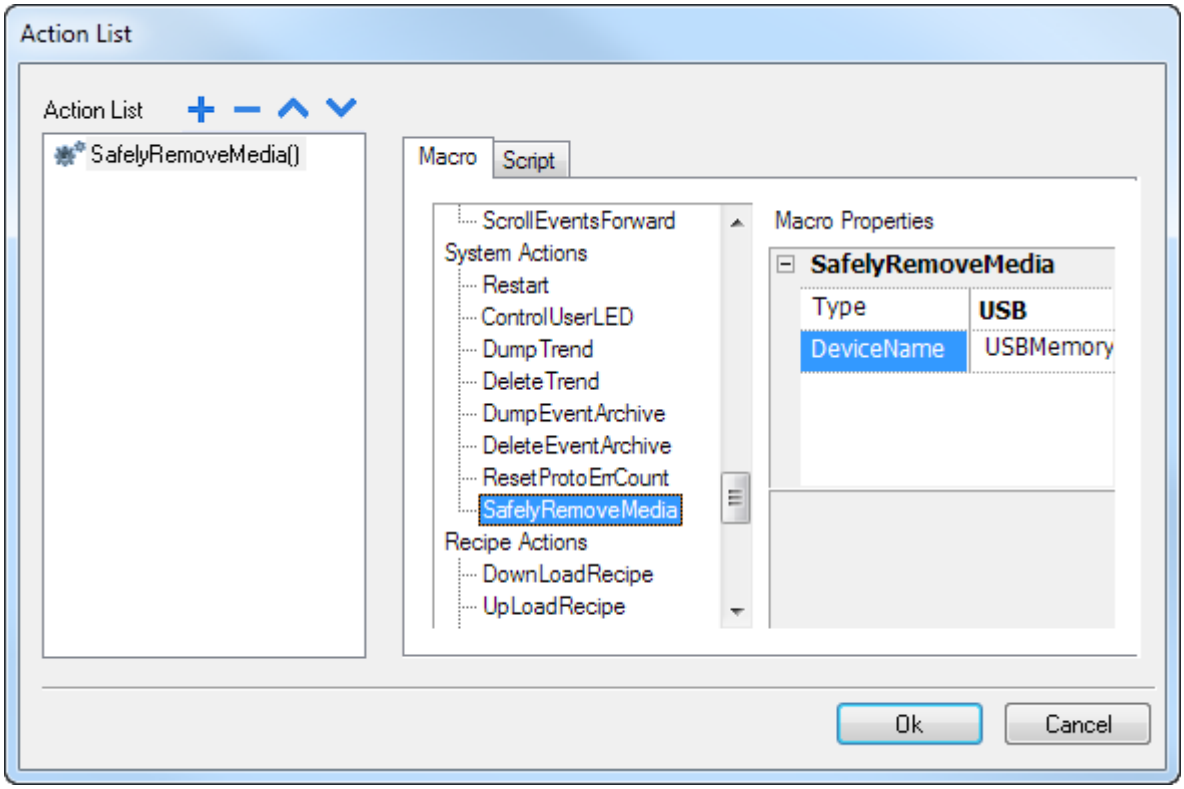


Figure 101

9.10.9 CopyCodesysProject

The action *CopyCodesysProject* copy the CODESYS 2.3 project files (.prg, ,chk and .sdb) located in the **source path** into the panel CODESYS folder. Files are automatically renamed to DEFAULT.CHK , DEFAULT.PRG , BOOT.SDB if it is of different name.
After copy of files the CODESYS module is stopped, reloaded and started again.

Parameters:

- **Source Path** path of project into external storage. Ex. \USBMemory\Codesys
- **Copy Symbols** if set to true copy also .SDB symbols file (used by CODESYS 2 ETH protocol).

To generate CODESYS project files, proceed as follow from CODESYS:
1) Execute Project -> Rebuild All (to generate an updated SDB symbol file)
2) Execute Online -> Create boot project (to generate the .CHK and .PRG files)

The *CopyCodesysProject* action can be used also to transfer a CODESYS project from one hmi panel to another. In this case files needed from hmi running CODESYS project are following:
- DEFAULT.CHK
- DEFAULT.PRG
- BOOT.SDB

Using System variables ***PLC -> PLC Status*** and ***PLC -> Get CopyCodesysProject Action Status*** the status of PLC and of *CopyCodesysProject* action can be verified.

9.11 Recipe Actions

The Recipe Actions macros are used in programming the recipe management.

9.11.1 DownLoadRecipe

The DownloadRecipe macro allows you to transfer a set of Recipe data to the controller. In macro properties, select the Recipe in the Recipe Name field and select the Recipe set you want to download. To download the currently selected Recipe set, select “curSet" in RecipeSet.

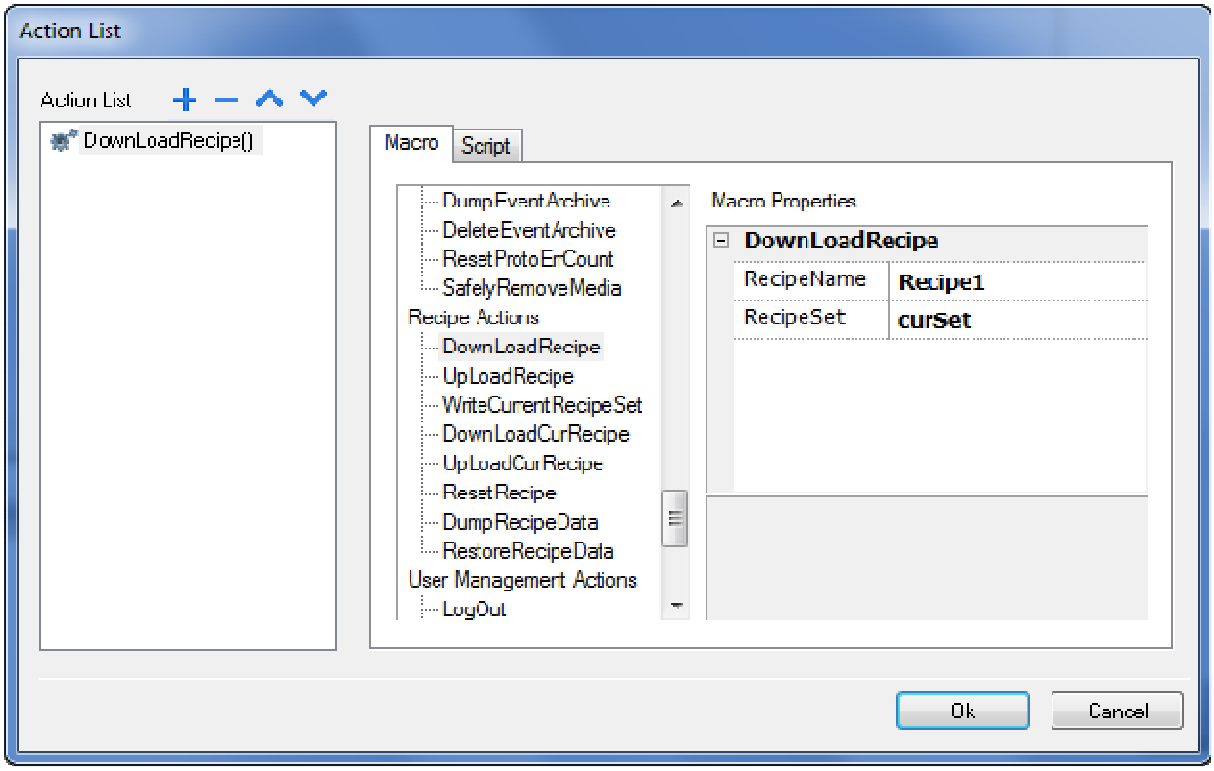


Figure 102

9.11.2 UpLoadRecipe

The UploadRecipe macro allows you to transfer the controller data to the Recipe set data. In the macro properties, select the Recipe in the Recipe Name and select the Recipe set that you want to upload. To upload to the currently selected Recipe set, select “curSet" in RecipeSet.

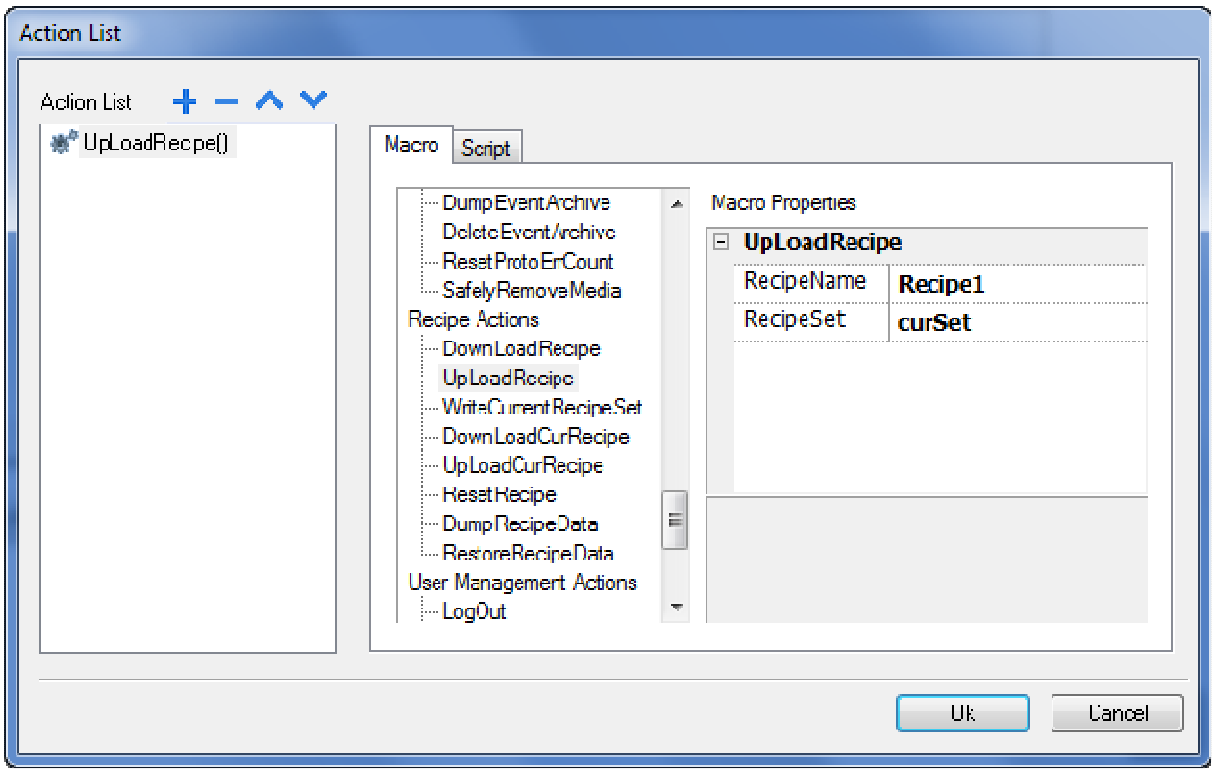


Figure 103

9.11.3 WriteCurrentRecipeSet

The WriteCurrentRecipeSet macro allows you to set the selected Recipe as current Recipe Set. In Macro Properties, select the Recipe and Recipe Set you want to set as the Current Recipe in runtime.

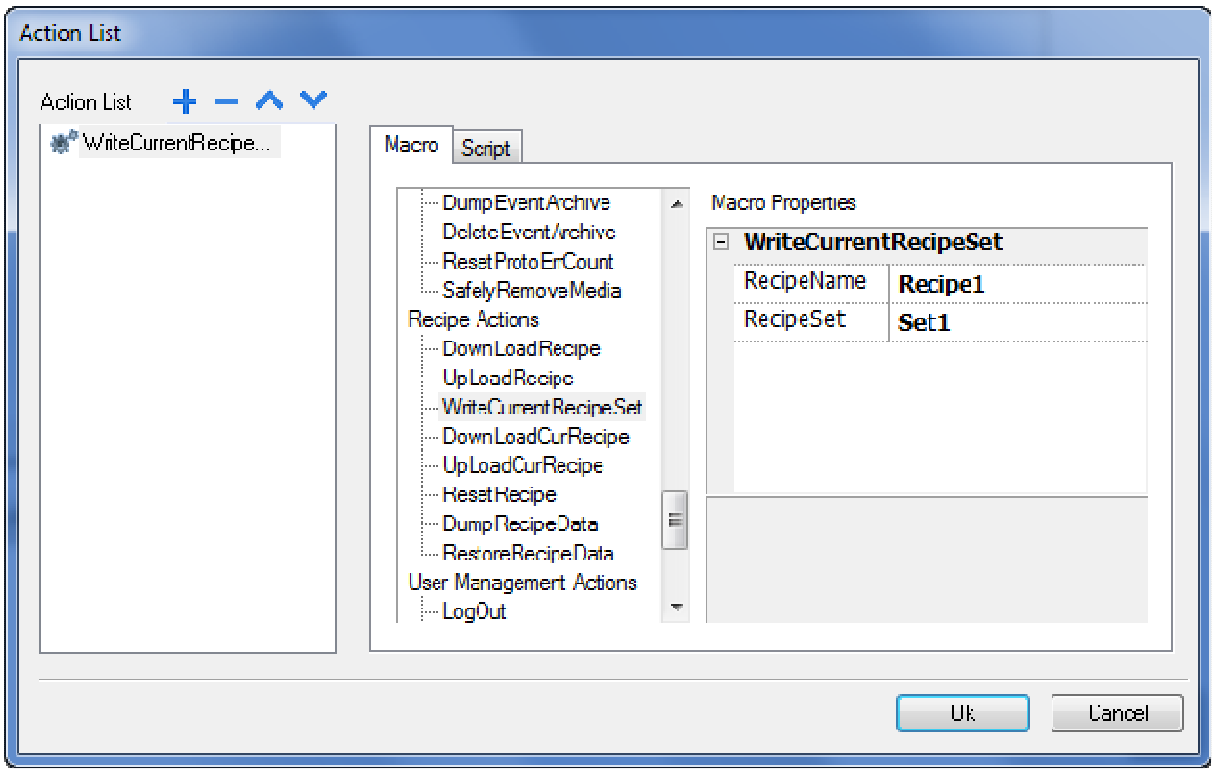


Figure 104

9.11.4 DownLoadCurRecipe

The DownloadCurRecipe macro allows you to transfer the current set of Recipe data to the controller. No parameter is required in the Macro Properties. This will download the currently selected Recipe and Recipe set to the controller.

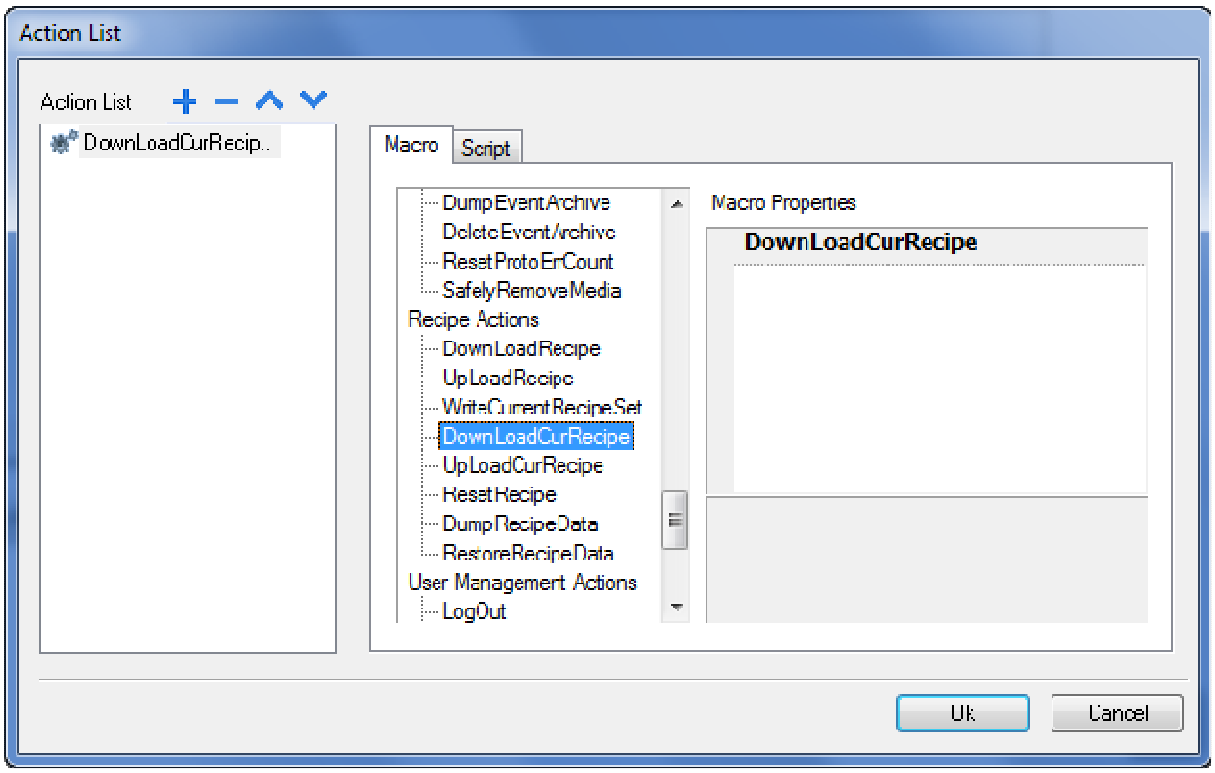


Figure 105

9.11.5 UploadCurRecipe

The UploadCurRecipe macro allows you to transfer the set of controller data values to a Recipes set. No parameter is required in the Macro Properties. This will upload the currently selected Recipe from the controller.

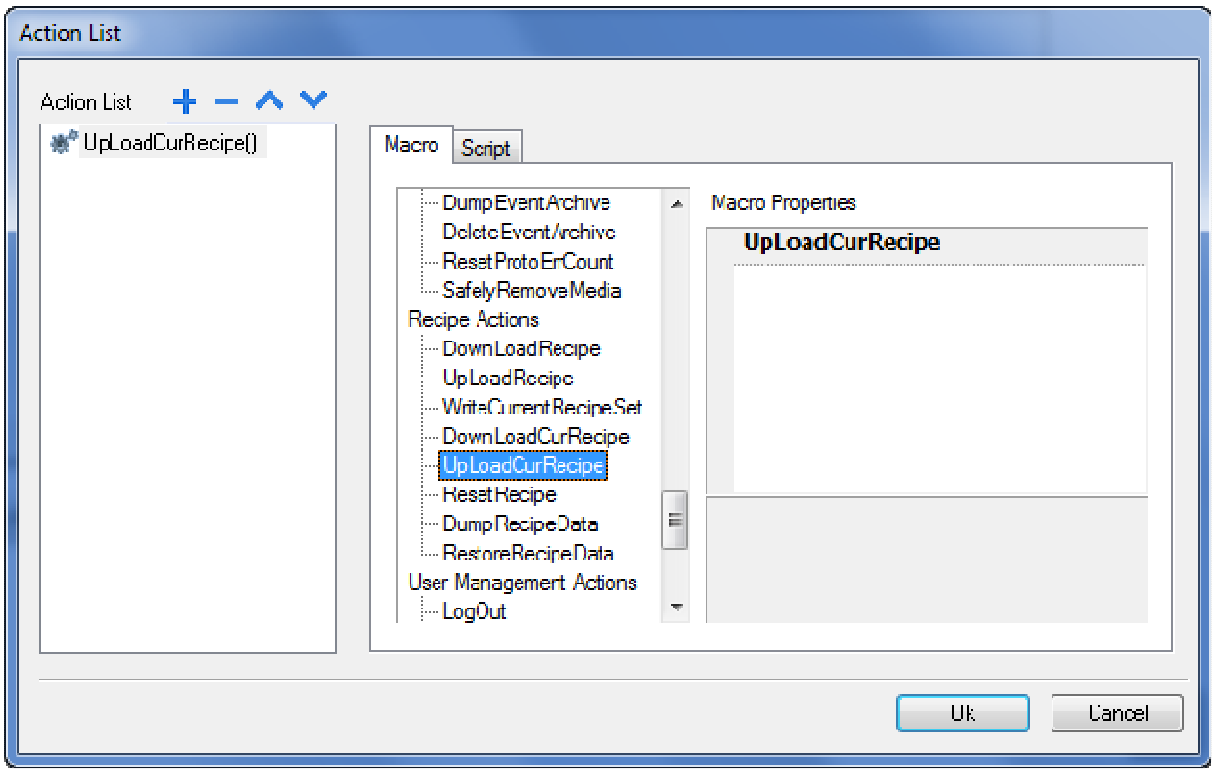


Figure 106

9.11.6 ResetRecipe

The ResetRecipe macro allows you to restore the factory settings for the Recipe data. The uploaded Recipes will be replaced with the original Recipe data. In the macro property, select the Recipe that you want to reset to factory settings.

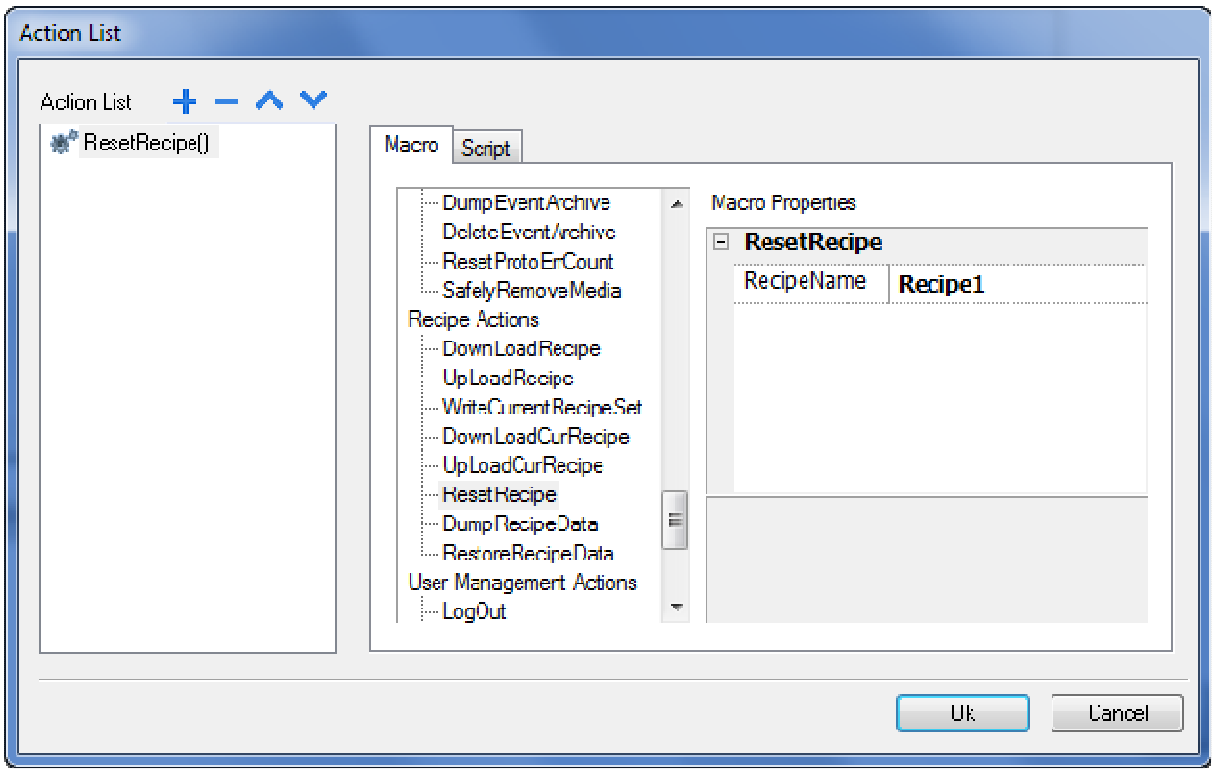


Figure 107

9.11.7 DumpRecipeData

The **DumpRecipeData** macro is used to dump recipes to internal or external storages. In the **Macro Properties**, define the location where to save the dumped file. Recipe data is saved in CSV format. Dump can be done in any external storage like USB, SD or network path.

NOTE *The external drives plugged in the USB port of the panel must have format FAT or FAT32. NTFS format is not supported.*

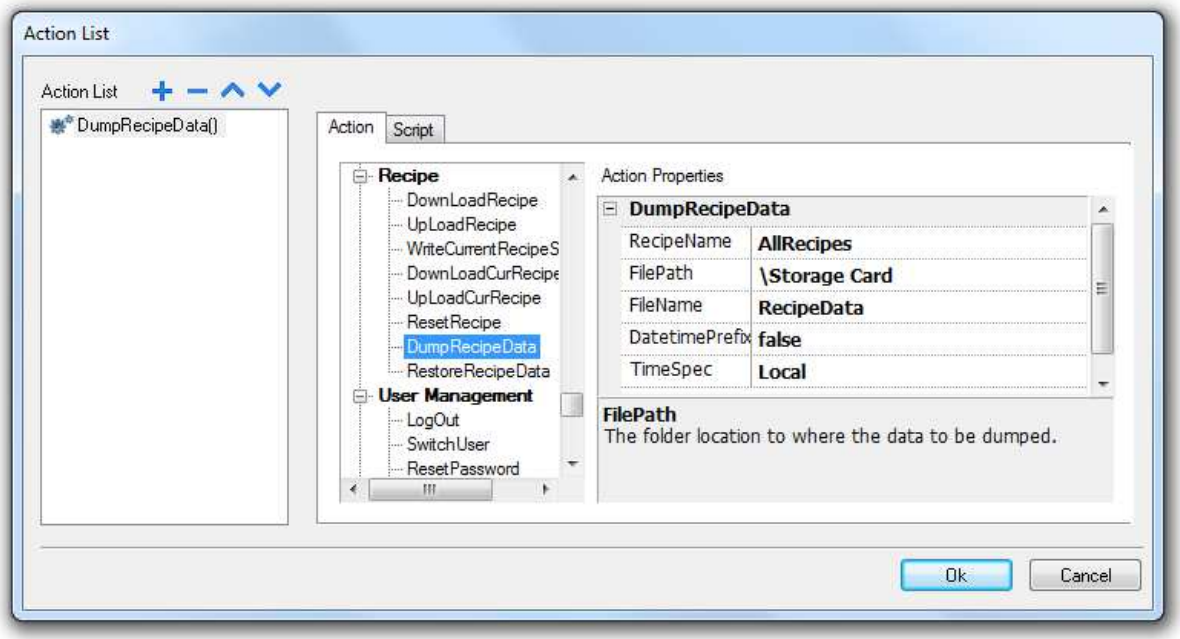


Figure 108

DateTimePrefixFileName
When this option is enabled the dumped file will have the Date and Time as prefix of the filename.
For example: if we are making a Dump at 10.10AM on 1-1-2012, then the file name will look like D2012_01_01_T10_10_recipe1.csv. [DYear_Month_day_THour_Minute_Filename]
This helps to know the Time at which the Dump was executed and also to identify which one is the latest.
TimeSpec define time format, **Local** for HMI time and **Global** for UTC time.

9.11.8 RestoreRecipeData

The **RestoreRecipeData** macro allows you to restore the Recipe data previously saved. In Macro Properties, provide the file full path of the Recipe files. Recipes to restore can be in any external storage like USB, SD or network paths.

NOTE *The external drives plugged on the USB port of the panel must have format FAT or FAT32. NTFS format is not supported.*

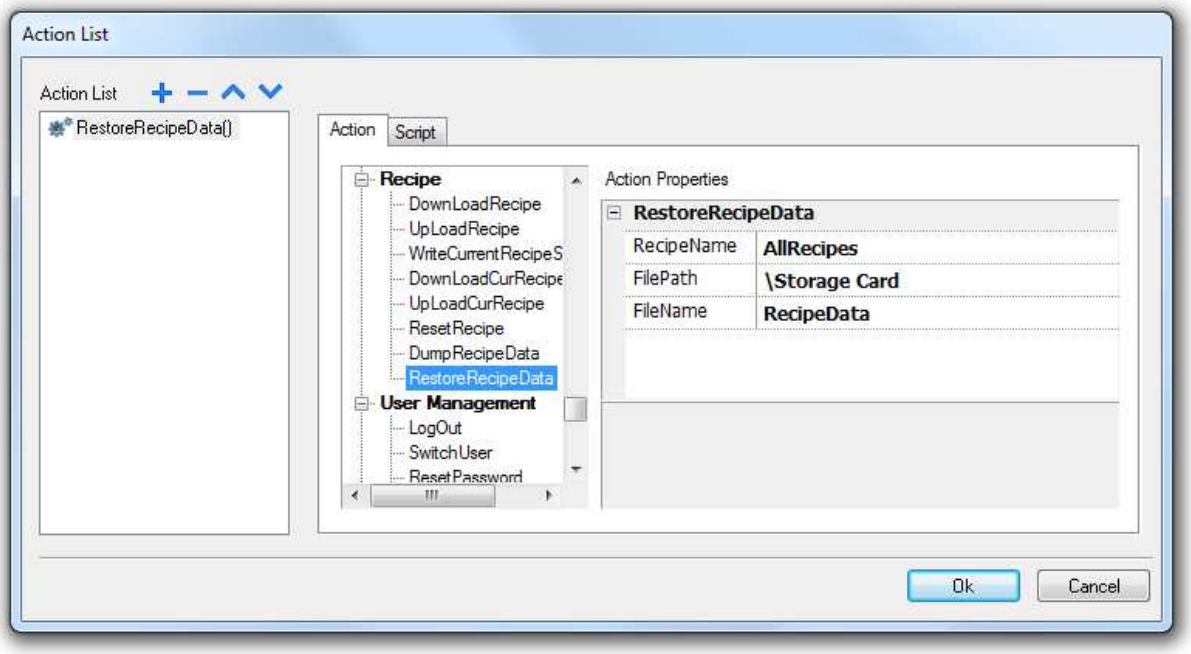


Figure 109

9.12 User Management Actions

User Management Actions macros have been designed for user management and security settings in Runtime.

9.12.1 LogOut

The LogOut macro allows you to log off the current user in Runtime. After executing the LogOut macro, the HMI behavior depends on whether a Default user is configured in the project or not.
If there is a Default user, the LogOut automatically logs in the Default user. If there is not a Default user or you logout from the Default user, then the log-in screen is shown.

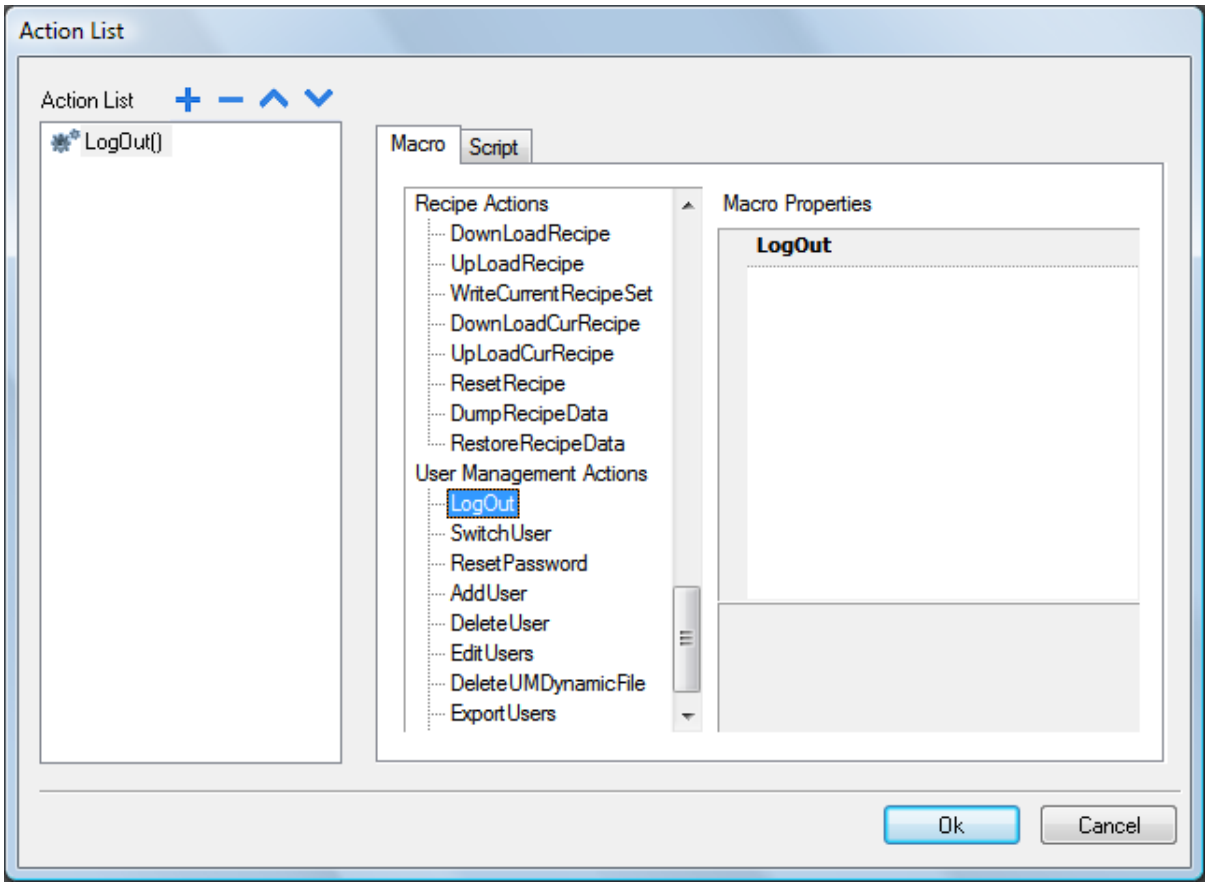


Figure 110

9.12.2 SwitchUser

The SwitchUser macro allows you to switch between two users without logging-out the logged-in user. The server continues running with the previously logged-in user, until the next user logs in. This means, after executing the SwitchUser macro, the runtime will display the User Login template. Internally, however, the server runs with the previously logged-in user. This action is useful for ensuring that there is always one user logged onto the system.

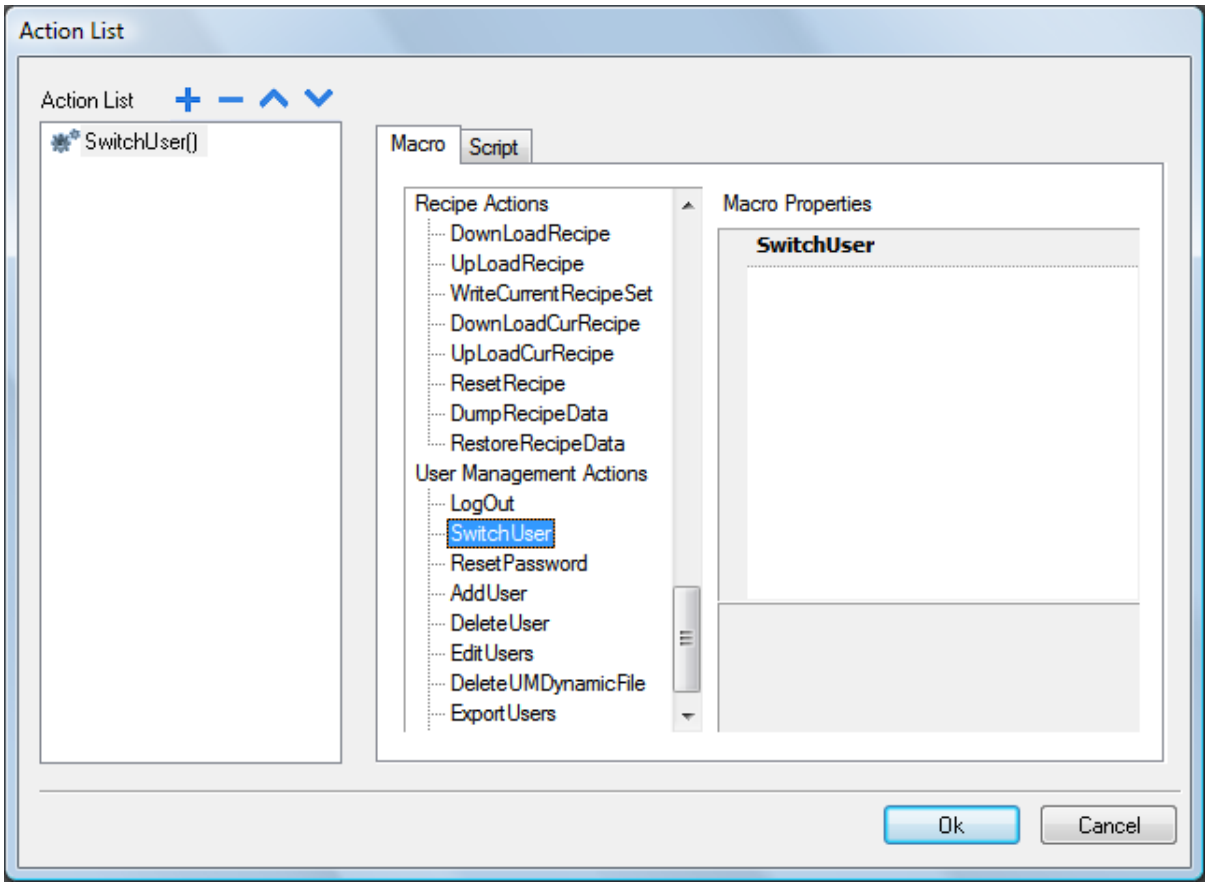


Figure 111

Click on the “Back" button to go back to the previously logged-in user.

Figure 112

9.12.3 ResetPassword

The ResetPassword macro allows the current user to restore his or her original password; this macro will restore settings specified in the project for the current user password. No parameter is required to set this macro.

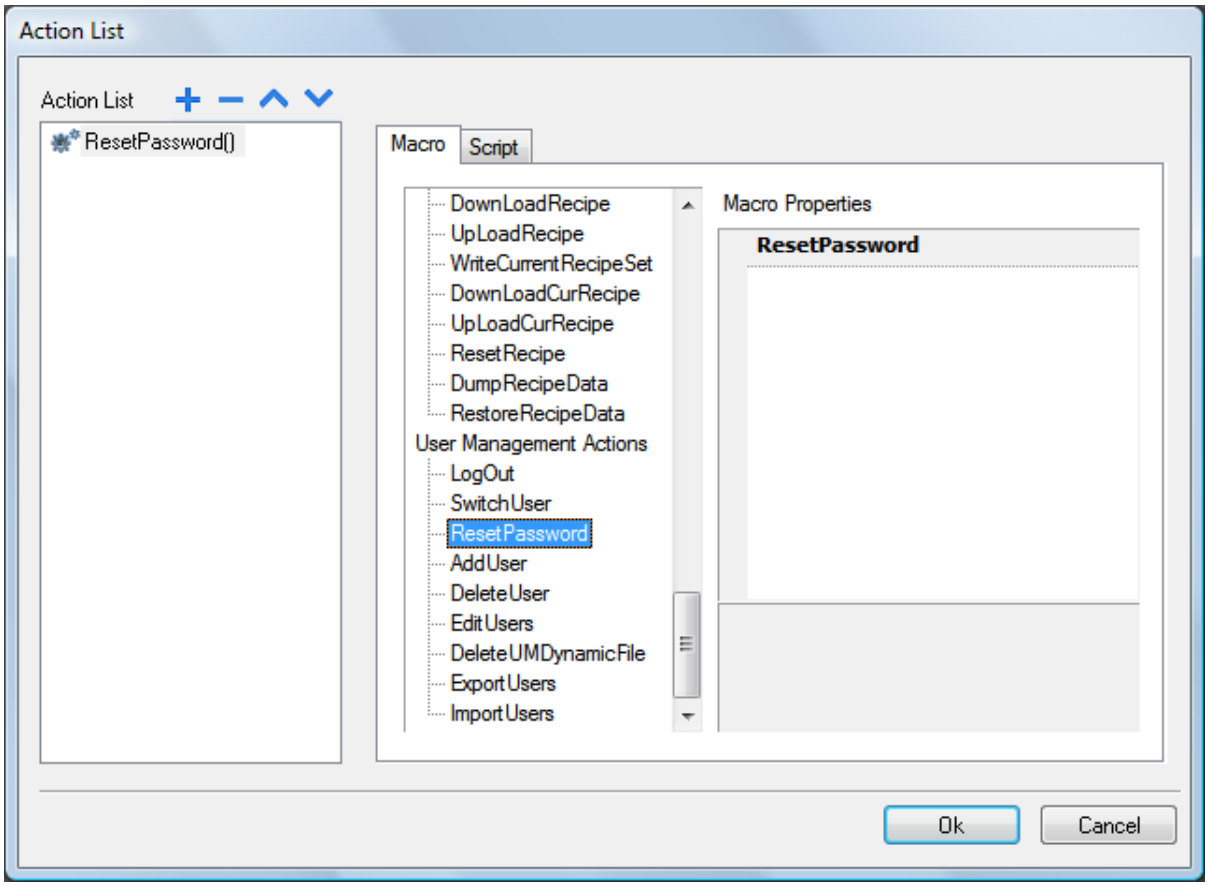


Figure 113

9.12.4 AddUser

The AddUser macro is used to add users at runtime. When this macro is executed, a template page pops up, where parameters for the user can be set. These parameters include Username, Password, Group, Comments, flags like ‘password must contain numbers’, ‘password must contain special character’, ‘user must change his initial password’, ‘enable logoff time’ and ‘Inactivity Logoff Time’. The User Log is shown in the figure below.

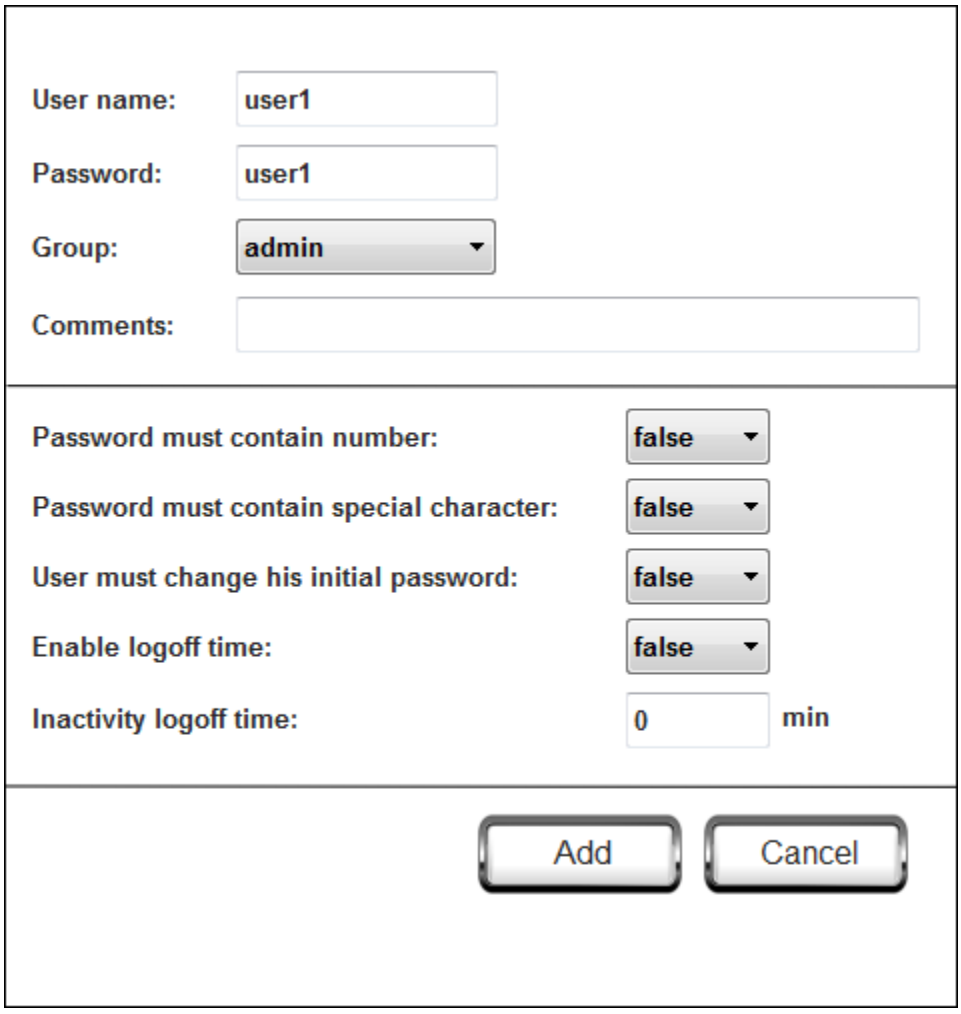


Figure 114

9.12.5 DeleteUser

The DeleteUser macro is used to delete users at runtime. Upon executing this macro, a template page will pop up where you can select the user you wish to delete. No parameters are required to set this macro. After executing the macro, the Delete User form will be displayed, as shown in the figure below.

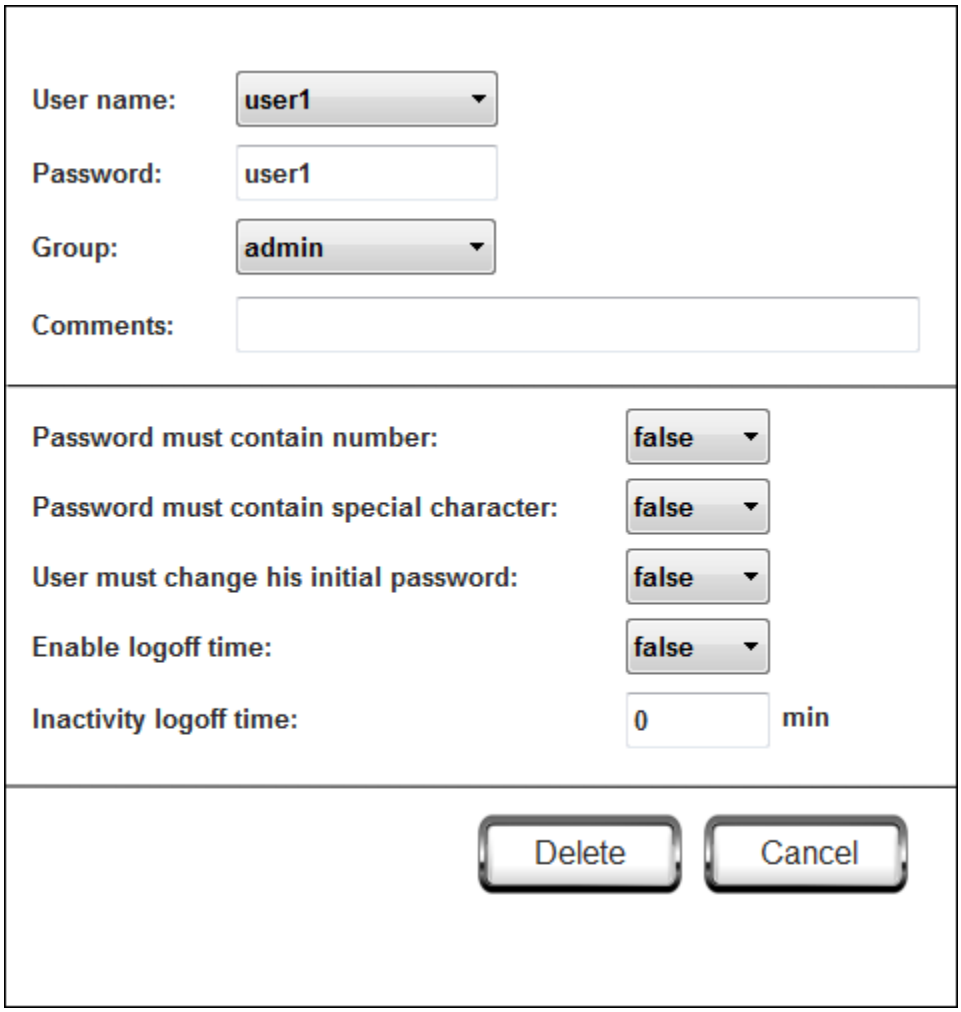


Figure 115

9.12.6 EditUsers

The EditUsers macro is used to edit users at runtime. When executing this macro, a template page pops up. Here you can select a user and modify this user's parameters (such as Username, Password, Group, Comments, flags like 'password must contain numbers', 'password must contain special character', 'user must change his initial password', 'enable logoff time' and Inactivity Logoff Time). After executing the macro, a User Edit form will pop up, as shown in the figure below.

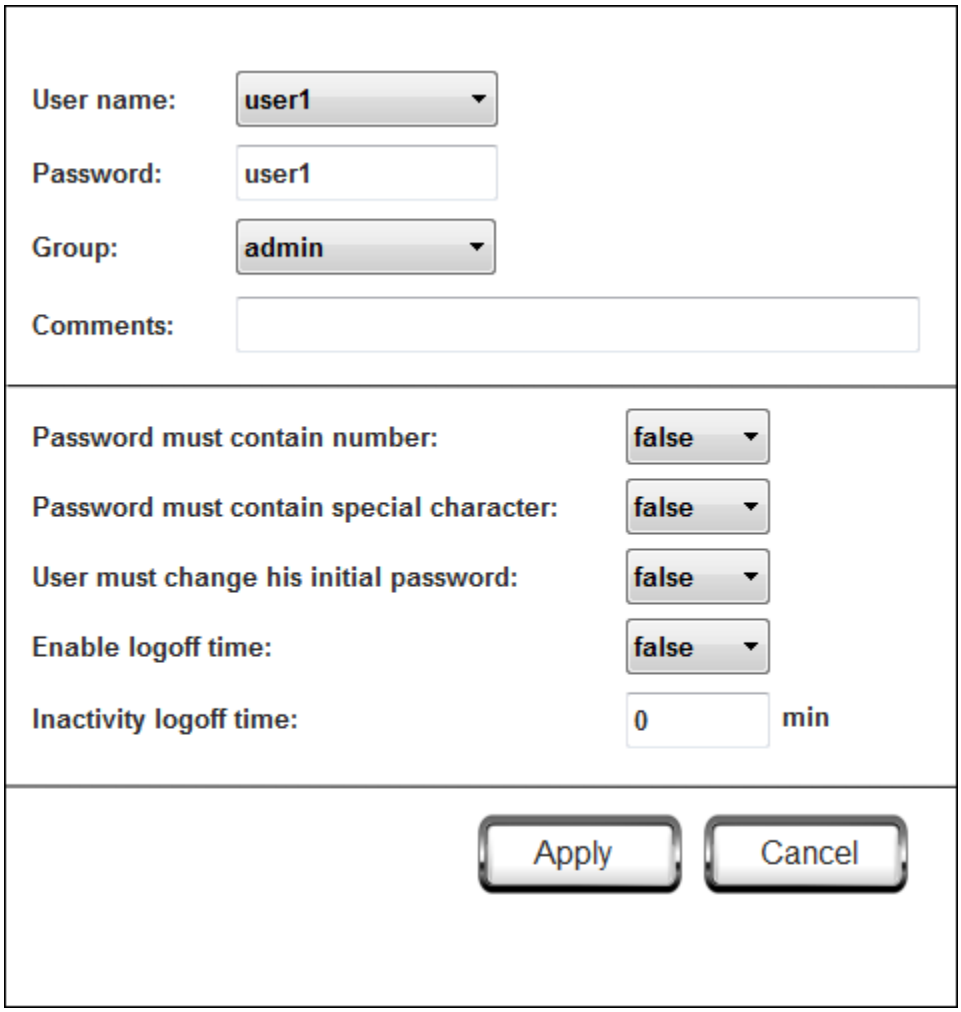


Figure 116

9.12.7 DeleteUMDynamicFile

The DeleteUMDynamicFile macro allows you to delete the dynamic user management file. This means that the users created, edited, or deleted in Runtime will be erased, and the server will restore the settings from the project, originally downloaded from JMobile Studio. No Macro Properties are required.

9.12.8 ExportUsers

The ExportUsers macro allows exporting user details to an xml file (usermgnt_user.xml). User details will be in encrypted form. In the Macro Properties, the destination folder path must be set to the location where the usermgnt_user.xml file is saved.
If using a USB drive plugged in to the USB port, the path will be “\USBMemory", followed by the specified folder in the memory (or left empty for root folder).

NOTE *The external drives plugged in the USB port of the panel must have format FAT or FAT32. NTFS format is not supported.*

Since the file is encrypted, there is no way to edit the user configuration from this exported file. This action is most useful for making a backup to be used for a later restore.

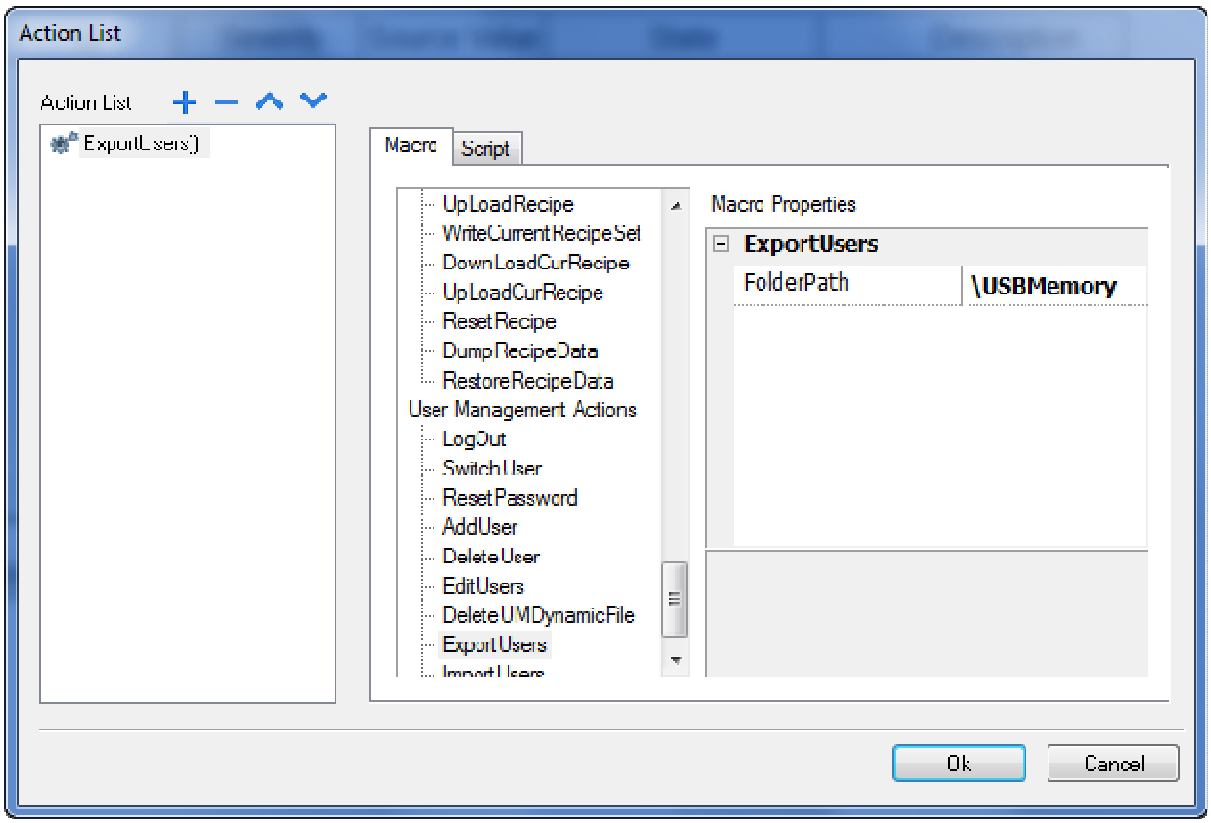


Figure 117

9.12.9 ImportUsers

The ImportUsers macro allows importing user details from an xml file named “usermgnt_user.xml". The path of the folder where the usermgnt_user.xml file is located must be specified in the Macro Properties.
If using a USB drive plugged into the USB port, the path will be “\USBMemory", followed by the specified folder in the memory (or left empty for root folder).

NOTE *The external drives plugged in the USB port of the panel must have format FAT or FAT32. NTFS format is not supported.*

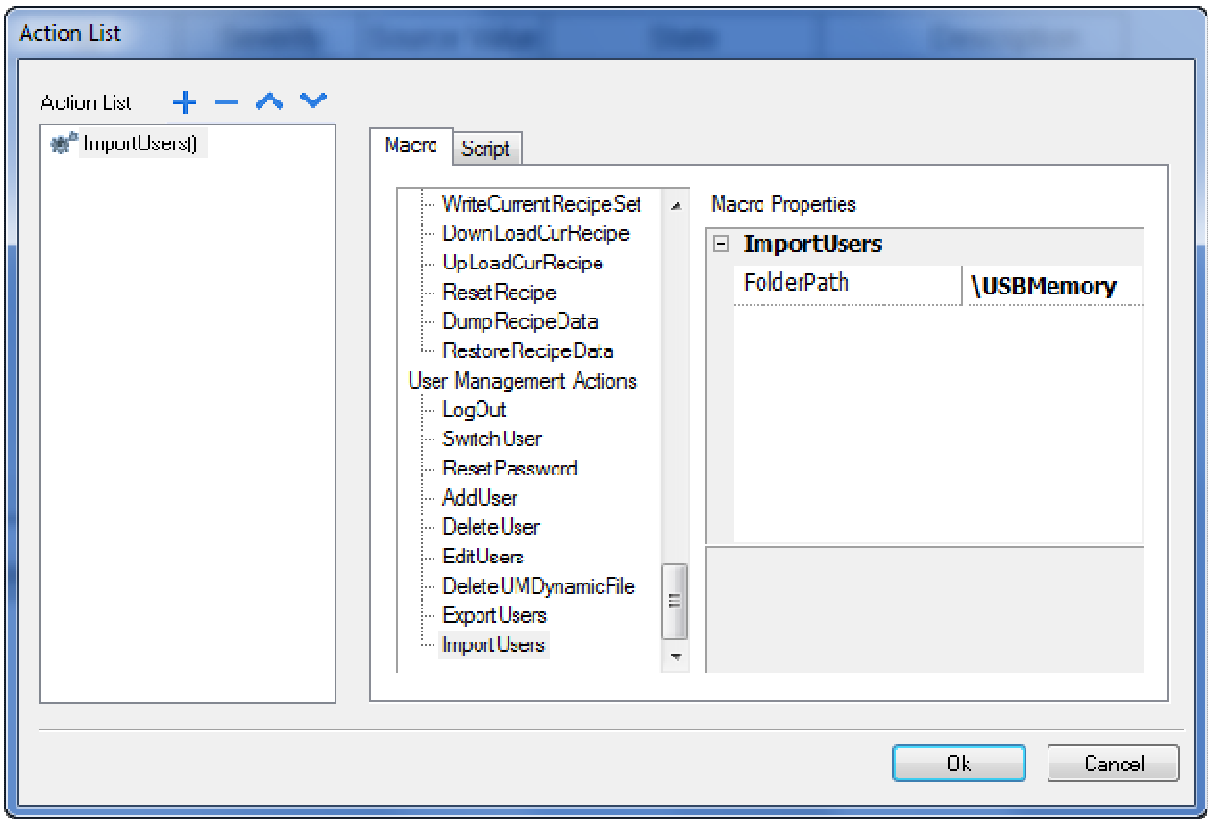

Action List
Action List
ImportUsers()
Macro
Script
WriteCurrentRecipeSet
DownLoadCurRecipe
UpLoadCurRecipe
ResetRecipe
DumpRecipeData
RestoreRecipeData
User Managemert Actions
LogOut
SwitchUser
ResetPassword
AddUser
DeleteUser
EditUsers
DeleteUMDynamicFile
ExportUsers
ImportUsers
Macro Properties
ImportUsers
FolderPath
\USBMemory
Ok
Cancel

Figure 118

9.13 Print Actions

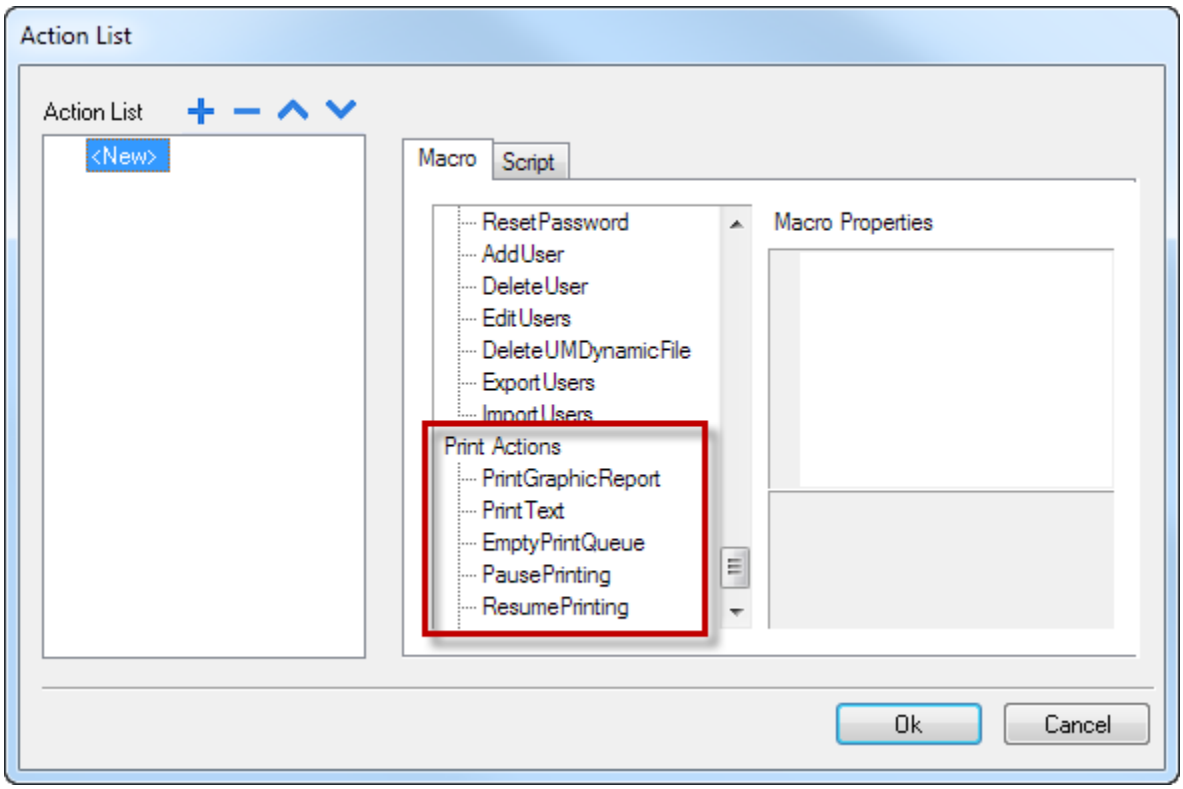


Figure 119

9.13.1 PrintGraphicReport

The **PrintGraphicReport** macro allows you to print a graphic report. You have to specify the report name in the combo box **reportName**. The option **silent** (default value is *true*), if set to *false*, allows you to open a dialog at runtime which asks the user to adjust printer properties.

9.13.2 PrintText

The **PrintText** macro allows you to print the string written in the field **text**. The option **silent** (default value is *true*) allows, if set to *false*, you to open a dialog at runtime which asks the user to adjust printer properties.

NOTE PrintText work in *line printing mode using a standard protocol common to all printers that support it. No custom drivers required for line printing.*

NOTE In *line printing, text is printed immediately line by line or after a timeout custom for each printer model (could take also minutes for some models not design for line printing).*

9.13.3 EmptyPrintQueue

The **EmptyPrintQueue** macro allows you to empty the current printing queue. If the macro is executed in the middle of the execution of a job, then the queue will be cleared at the end of the job.

9.13.4 PausePrinting

The **PausePrinting** macro allows you to put on hold the current printing queue. If the macro is executed in the middle of the execution of a job, then the queue is paused at the end of the job.

9.13.5 ResumePrinting

The **ResumePrinting** macro allows you to start the queue if previously it was put on hold.

9.13.6 AbortPrinting

The **AbortPrinting** macro allows you stop the execution of the current job and remove it from the queue. If the queue has another job, then, after aborting, the next one starts immediately.

10 Using JMobile Client

The JMobile Client provides remote access to the Runtime, and is included in the JMobile Suite installation. The JMobile Client consists of a simple standalone application; although it uses the same graphic rendering system as the server, it relies on a specified Runtime as Server for live data.

JMobile Clientfor Windows is available in the Runtime folder of the JMobile Suite root folder. Execute the JMobile Client application from the Runtime folder or from the start up menu (JMobile Suite- JMobile Client). The client will open in a browser-like style window. Type the server IP address (the panel's IP address) in the address bar (for example: http://192.168.1.12). The Client will connect to the server and the same graphical application running on the Server panel will be loaded in the client window.

JMobile Clientacts as a remote client and communicates to the server, sharing the local visualization with those Tag values that are maintained or updated by the communication protocol.

The HMI projects contain properties that let you know which page is currently displayed on the HMI and to force the HMI to switchto a specific page. These properties can be used to synchronize pages showed on the HMI and JMobile Client or to control an HMI with a PLC. Please refer toSyncOptionsfor more details.

NOTE *If any of the project files downloaded fromJMobile Studio to the hmi panel change for any reason (because corrupted or because has been manually changed via FTP client for example) an error message appears into theJMobile Client while downloading project. Error message appear like a "checksum does not match".*

10.1 The JMobile Client toolbar

The JMobile Client toolbar contain following:

- URL related to the hmi panel address
- A led indicating network requests. During data exchange the led becomes red.
- Reload button, useful for forcing project reload. Use "F5" to reload project from cache (simple reload of project) or "Shift" + "F5" to force a full clean redownload of project to the client (useful when project is changedin the target).
- Bookmark
- Settings

Figure 120

10.2 Settings &Time Zone Options

JMobile Clientprovides an additional option to handle the visualization of the timestamp information of a project and to configure client settings.

From the **Settings** dialog you now have access to a set of new options:

- **Update Rate** (default 1 second): polling frequency used by JMobile Client to synchronize data from server.
- **Time Out** (default 5 seconds): max time wait from JMobile Client before considering a request lost and repeating it.

- **Reuse connection:** allow to reuse the same TCP connection for multiple HTTP requests reducing network traffic and latency. Please note that in some circumstances, enabling this option can lead to the opposite result: very high latency. In particular this can happen when connecting through a HTTP proxy, like often seen in 3G connections, if proxy server is not immediately terminating old requests (causing server to saturate the available connection sockets).
- **Enable compression:** enabling this flag can improve download times on very slow connections. For LAN ones, the cpu overhead required to accomplish compression server side is just too high. So by default the flag is disabled.
- **Time Settings**:information is used by the client to adapt the widget timestamp information according to the desired behavior. Options for Time Settings:

Use Widget Defaults	For each widget use time information according to the widget settings provided at the time of programming.
Local Time	Translates all timestamps used in the project into the PC local time where the client is installed.
Global Time	Translates all timestamps used in the project into UTC format.
Server Time	Translates all timestamps used in the project into the same used by HMI panel/server in order to show the same time.

NOTE *This feature requires you to set the HMI RTC with the correct time zone and DST (Daylight Savings Time) options.*

10.3 Workspace

Using JMobile Client, project files will be uploaded from panel and placed into a cache directory.
Cache folder where project are temporary stored is under:

- *%appdata%\Exor\[build number]\client\cache*

where [build number] is a folder named as build number like *01.90.00.608.*

11 Using the Integrated FTP Server

The HMI Runtime system features an integrated FTP server that can be used to get access to the internal flash disk data.

NOTE *Folders present on the Flash disk external to the runtime directory are not accessible via FTP; external USB drive and SD Storage Card are not accessible via FTP.*

You can use any standard FTP client program to connect to the panel FTP server. The FTP server responds to the standard port 21 when using the IP address assigned to the panel as host.

NOTE *The server supports only ONE connection at a time; if you are using an FTP client which is configured to multiply the connections to the server in order to speed up the transfer operation, you will need to disable this feature in the client program or set the maximum number of connections per session to 1.*

The FTP server is configured by default to accept incoming connection from the following account (when User Managemnt/Security is disable):

- User name: *admin*
- Password: *admin*

FTP permissions and account information can be changed from the “UserGroups” under the “Security” item of the project folder as shown in the following figure.

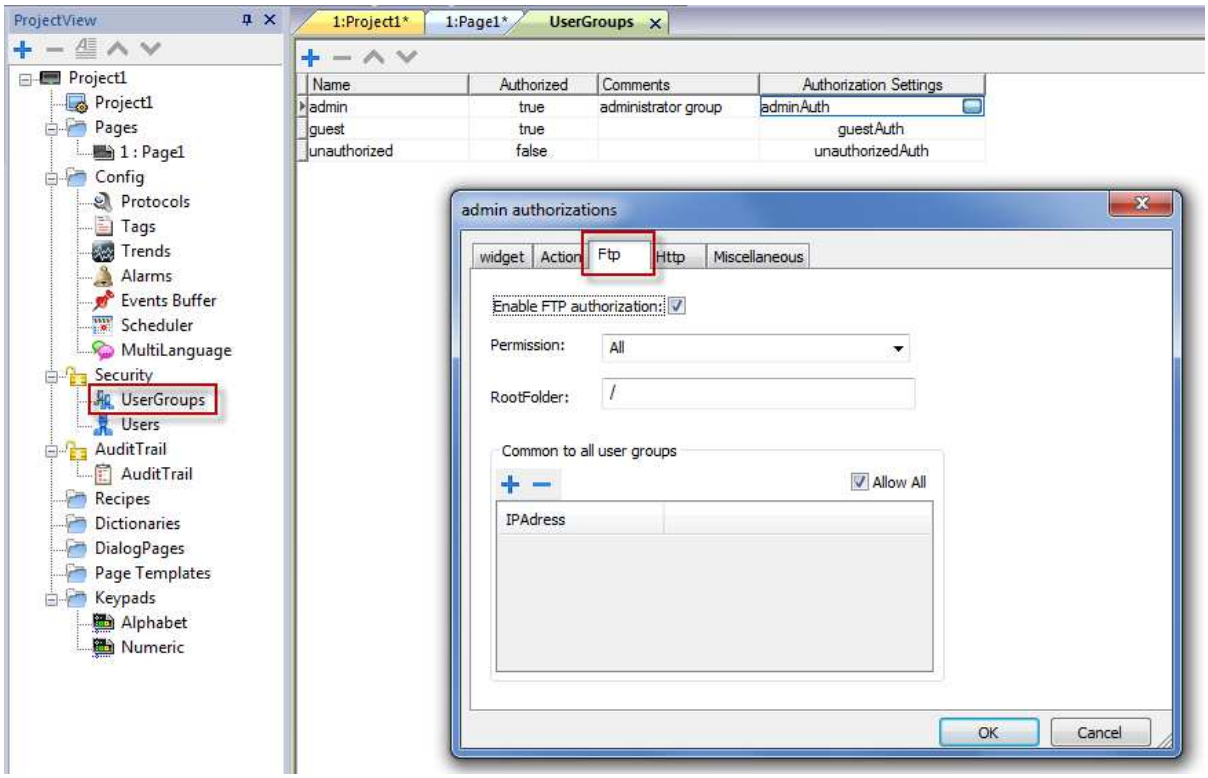


Figure 121

Additional information can be found later in this document in the chapter titled “FTP Authorizations”.

12 Using ActiveX Client for Internet Explorer

In the standard distribution of JMobile Suite, a JMobile Client and an ActiveX Client is provided. ActiveX components are NOT installed by default to the Target devices, in order to save space in the flash memory.

12.1 Installing ActiveX

The ActiveX component is distributed with the JMobile Suite installation package. The related files are located in the Runtime folder of the JMobile Suite installation directory. The files, “HMIAX.cab" and “HMIClientAX.html", should be copied into the workspace folder of the Target device, where the Runtime is installed. The file copy can be done using the panel FTP server.

Starting from v1.90 (b608) of JMobile Suite has been introduced software **plug-in** support (ref. to chapter on software Plug-ins chapter for more details) to simplify ActiveX installation. Just enable ActiveX plug-in from project properties and install/update runtime to add ActiveX files to the runtime and transfer it into the target without the need of manual copy of it via FTP.

NOTE *This ActiveX requires Microsoft Visual C++ 2008 Redistributable Package (x86) installed on your system. You may need to download the Microsoft Visual C++ 2008 Redistributable Package (x86) from the Microsoft web site.*

NOTE The ActiveX plug-in require about 10MB of space. Enable it only if required by the HMI application to keep the smallest footprint for the application.

12.2 HTTP Access to ActiveX files

When security is enabled, ActiveX files “HMIAX.cab” and “HMIClientAX.html” have tobe accessible from the http server embedded into the runtime. Refer to **HTTP Authorizations** chapter for more details.

12.3 Internet Explorer Settings

Internet Explorer settings must be changed adding the panel’s IP to the list of the trusted sites.
In Tools – Internet Option Security tab choose “Trusted sites". Then click on the “Sites" button. Type in the IP address of the Target device the location where the ActiveX component has been installed and it will be loaded to the browser.

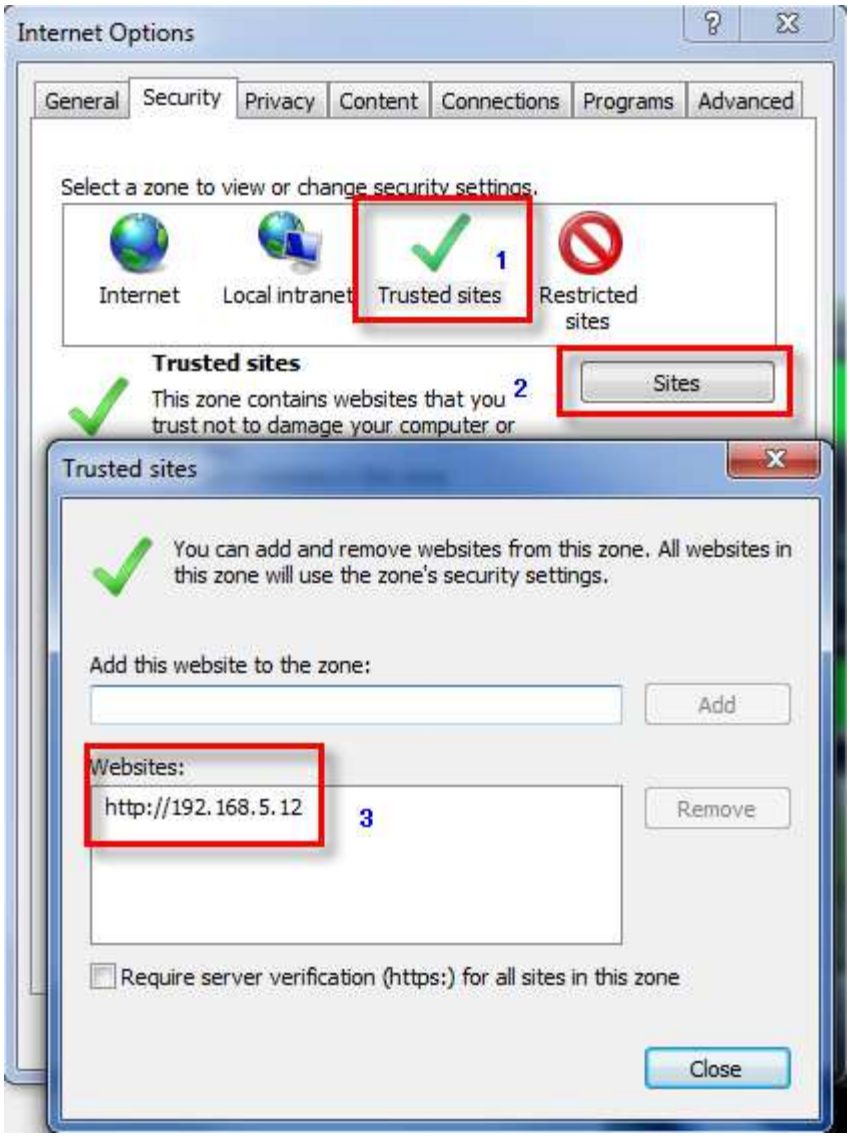


Figure 122

12.4 Security Setting for Trusted Site Zone

Set your Internet Explorer Browser as seen in the following images:

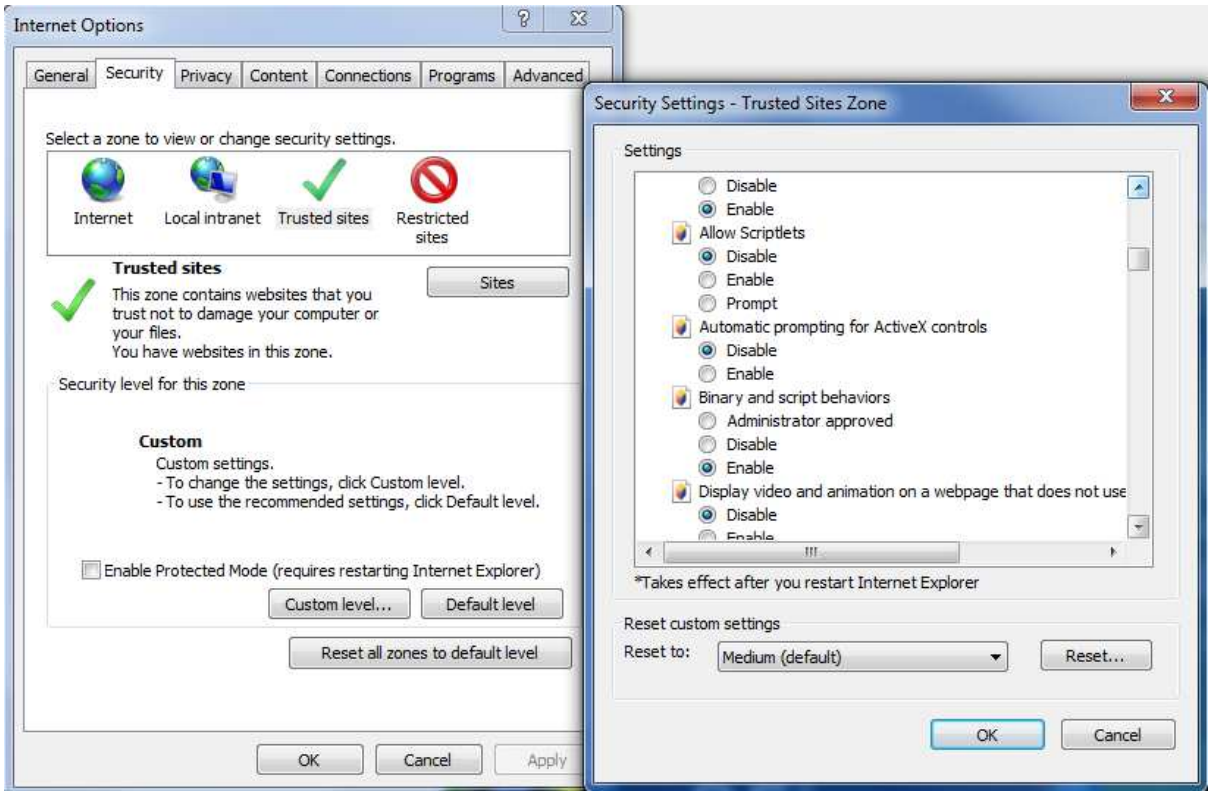


Figure 123

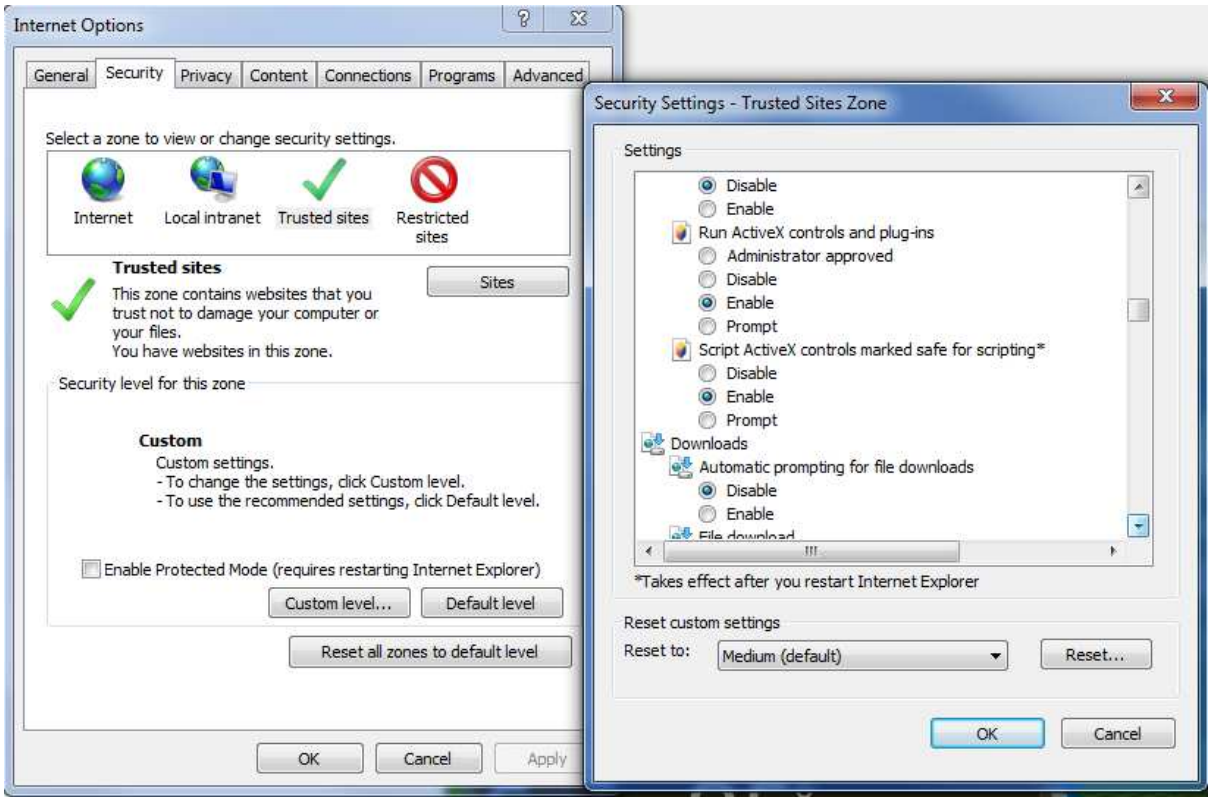


Figure 124

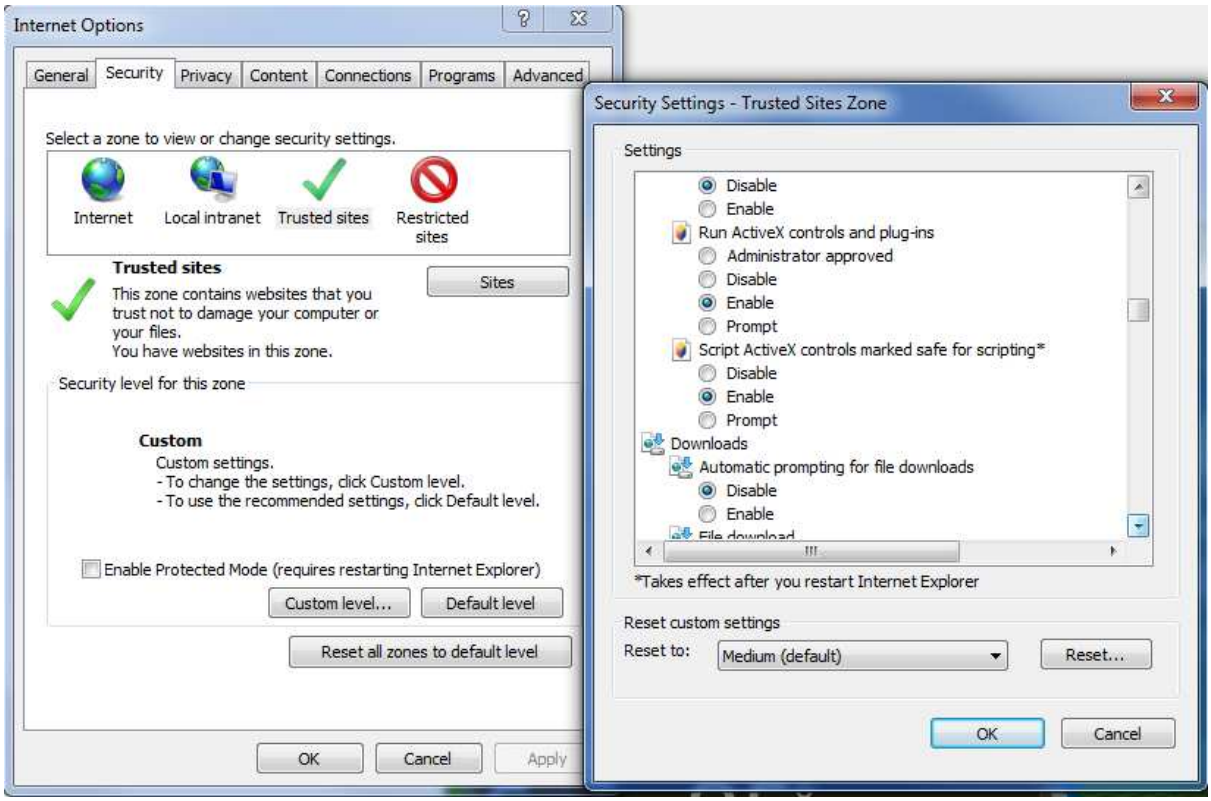


Figure 125

12.5 Install Active X in Internet Explorer

In Internet Explorer, allow the installation of the ActiveX component when the question pops up in your browser.

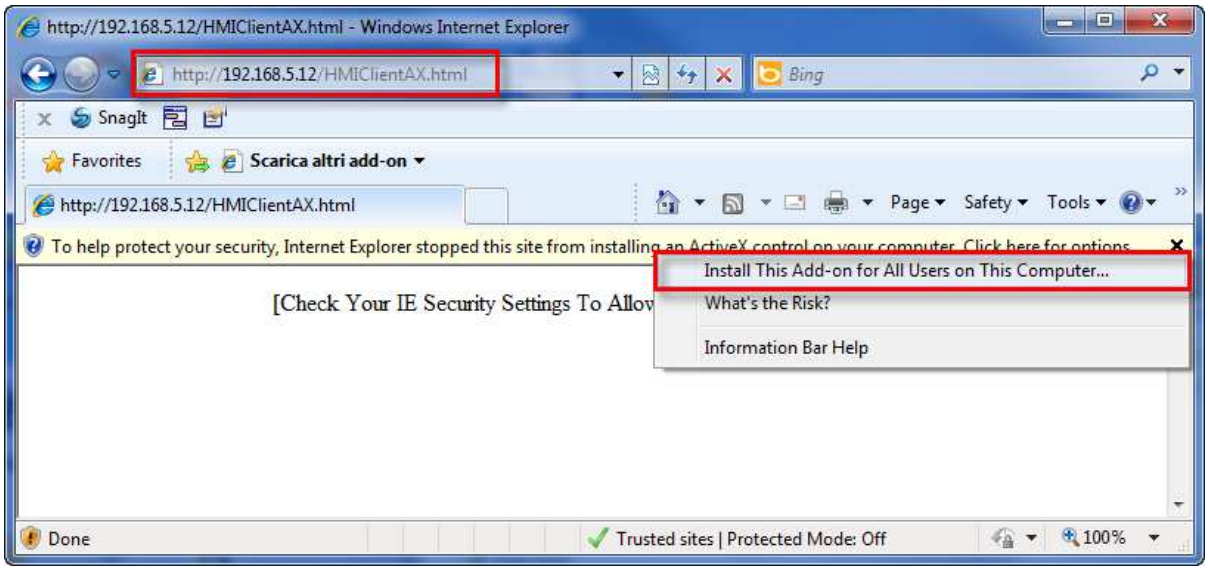


Figure 126

In case you are using a Vista or Windows 7 operating system, you need to click on Yes on User Account Control, as shown in the following picture.

User Account Control
Do you want to allow the following program from an unknown publisher to make changes to this computer?
Program name: Unknow Program
Publisher: Unknown
File origin: Download from the Internet
Show details
Yes
No
Change when these notifications appear

Figure 127

12.6 Uninstalling Active X

To remove the ActiveX component from your system, you must delete it from the computer. By default, the component is installed in the following folder:
C:\Program Files\Exor\HMIClientAX

12.7 ActiveX information

The ActiveX is able to show projects at a maximum pixel resolution of 1200 x 800.

13 Using VNC for Remote Access

VNC is a software for remote control. With VNC, you can see the HMI application remotely and control it with your local mouse and keyboard, just like you would if you were in front of it.

VNC is useful for administration and technical support. To be used it requires that a server is started on the HMI device; a viewer is used for connecting from a remote location.

13.1 VNC Server

Starting from v1.90 (b608), the VNC Server has been added as plugin (ref. to Plugins chapter for more details) to allow developers of hmi applications to choose if enable & download it as part of the runtime. Just enable it from project properties -> plugins and install/update runtime to download it into the target.

VNC server is located in folder **\Flash\qthmi\VNC** and can be activated using macro **launchVNC.**

LaunchVNC macro is used to open the VNC configuration dialog. From the configuration dialog you can:

- **Start / Stop / Restart VNC Server** in **Control** Tab
- Enable **security** and set **password** in **Options** tab that will be used later for access using a VNC viewer.
- Enable **Autostart** in **Advanced** tab to activate VNC server automatically every time the HMI panel start.
    - **Silent Startup** (usefull only when Autostart is enabled) prevents the VNC dialog from appearing at panel start-up and keeps it open in the background.

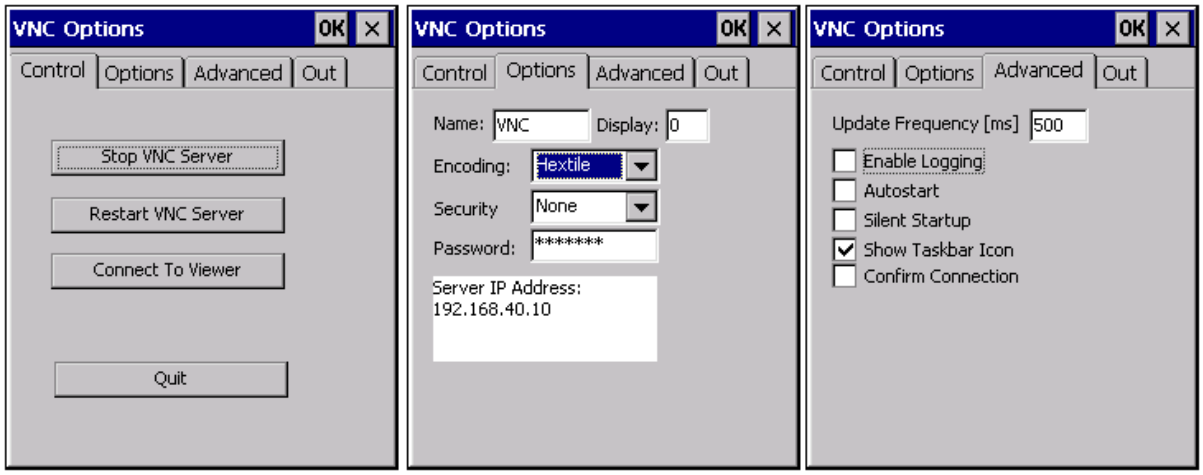


Figure 128

OK button on top/right of VNC server configuration dialog is used to confirm and save changes.

Advanced configurations are provided for expert users when VNC server is used in conjunction with a VNC repeater to bypass firewall problems or to optimize VNC performances based on network configuration.

NOTE The *VNC server uses port 5900/TCP.*

NOTE The **Password** of VNC server is null as default. Password can be changed via VNC viewer or with an external usb keyboard attach to the hmi panel.

NOTE For developers, a macro **LaunchVNC** is available in **Developer tools**.

NOTE **Show Taskbar icon** flag is used by tech support when debugging problems out of KIOSK mode. This flag is not usefull for standard hmi users.

NOTE The VNC Server has been design for embedded HMI panels WCE based. Win32 platform not supported for VNC Server.

NOTE The VNC Server allows only one single client. Two or more connected in the sametime is not allowed.

NOTE Drag and Drop of Windows is not supported yet by VNC server.

13.2 VNC Viewer

A VNC viewer is not provided as part of JMobile Suite. However, many types of VNC viewers are freely available. One example of compatible VNC viewer is TightVNC.

14 Alarms

The Alarm handling has been designed to provide alerts through pop-up messages, typically to issue warnings, to indicate any abnormal conditions or any malfunctions in the system under control. Whenever a Bit goes high, or the value of a Tag crosses the limit of deviation defined in the Alarm configuration, the respective Alarm message(s) will be displayed in a special dialog. Or, alternatively, you can program certain macro actions to be executed when the Alarm is triggered.
Please note that, in JMobile Studio, there is no default action associated with a triggered Alarm. The visualization of a specific page containing the Alarm Widget is optional, and the specific action executed when the trigger condition is verified can be any one of the actions found on the Action list.

The configuration of an Alarm determines whether or not the alarm requires user acknowledgement. It can also be used to determine how the Alarm appears when displayed on the HMI device (like background and foreground color). Alarm Configuration also determines whether, and when, the corresponding alarm is logged to the Event list.

For Alarms displaying critical or hazardous operating and process status, a stipulation can be made requiring the plant operator to acknowledge the Alarm.

The Alarms are configured in the alarm manager and, thus, are a component of all screens of a project. More than one Alarm can be displayed simultaneously in the alarm widget, depending on its configured size. An Event can trigger the closing and reopening of the Alarm window.

Please note that, in JMobile Studio, working with Alarms is similar to working with Events. In general, there is no absolute need to have a pop-up dialog when an Alarm is triggered. Any “background" action (from the list of available actions) can be associated with this Event.

14.1 Alarm Configuration Editor.

In the Project Workspace, double click on Alarms to open the Editor. Then add the Alarms by clicking the “+" button.

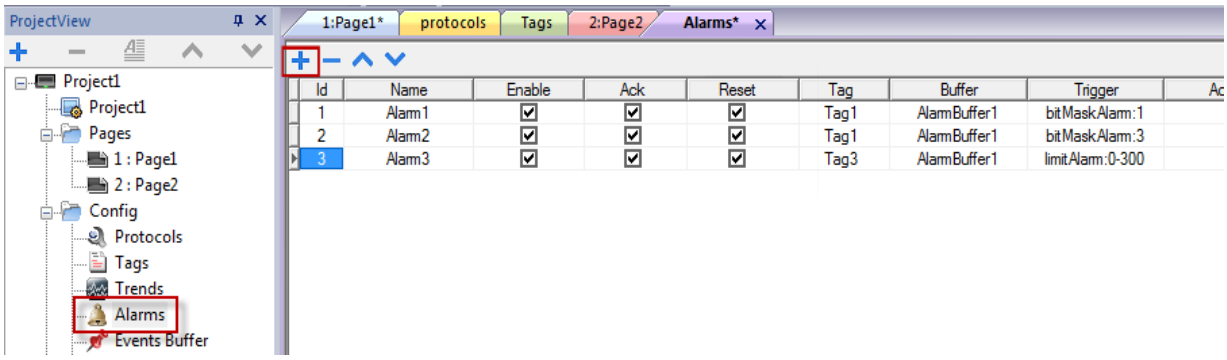


Figure 129

Name
Specifies the name of the Alarm.

Enable
A user can enable or disable the triggering of particular alarms. Alarms can be enabled or disabled on Runtime as well (for more information, please see Chapter Enable / Disable Alarms in Run-time).

Acknowledgement
For an alarm that needs to be acknowledged by the operator (when the alarm is triggered), select the check box to enable the Acknowledgment. If checked, an operator is required to acknowledge this alarm any time it is triggered, before it will be cleared from the Active alarm widget.

Reset
This check box, specific to each alarm, works in conjunction with the acknowledge check box. After an alarm requiring acknowledgment has been acknowledged, it will be cleared from the alarm list. If the Reset check box is checked, the alarm will continue to be listed in the alarm list, as “Not Triggered Acked”, until the Reset button present in the alarm widget is pressed.

Buffer
Specifies the Buffer file to which the Alarm history will be saved.

Trigger
This selection determines the triggering condition for an alarm.
Three Alarm types are available:

- **Limit Alarm** A Limit Alarm is triggered when the monitored Tag value goes OUTSIDE of its given boundaries (low limit and high limit). When the Tag value is equal to its low or high limit, the alarm is not triggered.

- **Bitmask Alarm** To get a valid trigger, the bitwise AND operator compares each bit of the bitmask with the Tag value corresponding to that Alarm. If both bits are on, the alarm is set to true. When the Bitmask Alarm is selected, you can specify one or more Bit positions inside the Tag. When one of the Bits is set, the alarm is triggered. The Bit position must be given in decimal format; if more Bits are specified, each position must be separated by a “,".
  Bitmask is a position, so it starts from zero (0).

- **Deviation Alarm** For the Deviation Alarm, a predefined “set point", as well as a value for “deviation" will be given. If the percentage of deviation of the Tag value from the set point exceeds this deviation, then the trigger condition becomes true.

$$|Value_{now} - SetPoint| > \left(\frac{deviation}{100} \times SetPoint\right)$$

Tag
Attach the Tag for which the Alarm shall periodically check the Tag value, so that the respective alarm(s) is triggered when this deviates from its limits. (The Alarm function will refer to the value of this Tag, or to the state of a Bit, in the case of Bitmask, to determine when to trigger the Alarm.)

Actions
Define the action(s) to be executed for the specific Alarm. Actions are executed by default when the specified trigger condition becomes true. Additional conditions can be specified in the “Events" configuration (in the last column of the Alarm editor, as explained in the chapter Action Enable).

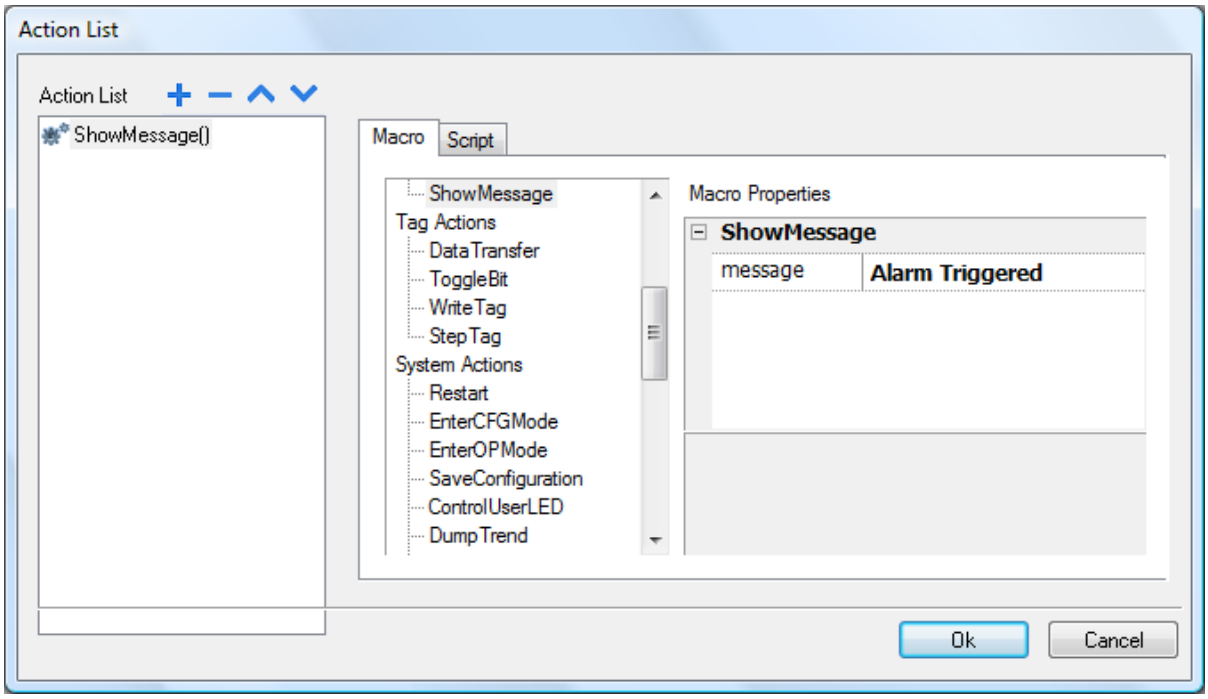


Figure 130

Description
This is the description of the alarm. The Alarm description is normally text; this text supports the multiple language features. The text can be a combination between static and dynamic parts, where the dynamic portion includes one or more tag values. Please see the chapter "Live Data in Alarm Description" for further information about this feature.

Color
Foreground and Background colors of alarm rows (Active alarms widget) can be applied based on the status of alarm (ex. Triggered, Triggered Ack etc).

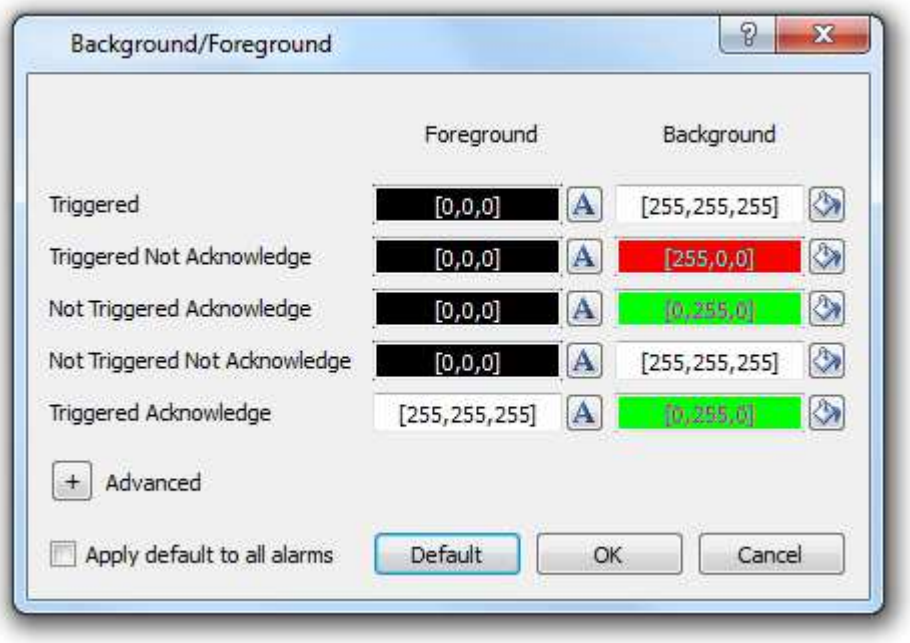


Figure 131

AckBlink
Make alarm row (of Active alarms widget) blink when an alarm is triggered. Stop blink when alarm has been ack-ed. Blink can be used only with alarms that have the Ack flag enabled.

Severity
A user can indicate the Severity of each alarm. If multiple Alarms are triggered simultaneously, the actions will be executed based on Severity settings.

Events
These options allow you to specify conditions relating to the following matters: when the Alarms events are to be logged, when the Alarms Widget View is to be refreshed or updated by the system, and some particular options for action execution. Setting Events is described in a dedicated chapter.

14.2 Alarms' State Machine

The HMI system implements an alarm State Machine which is described by the following figure.
The graph includes states and transitions between them according to the selected options and desired behavior.

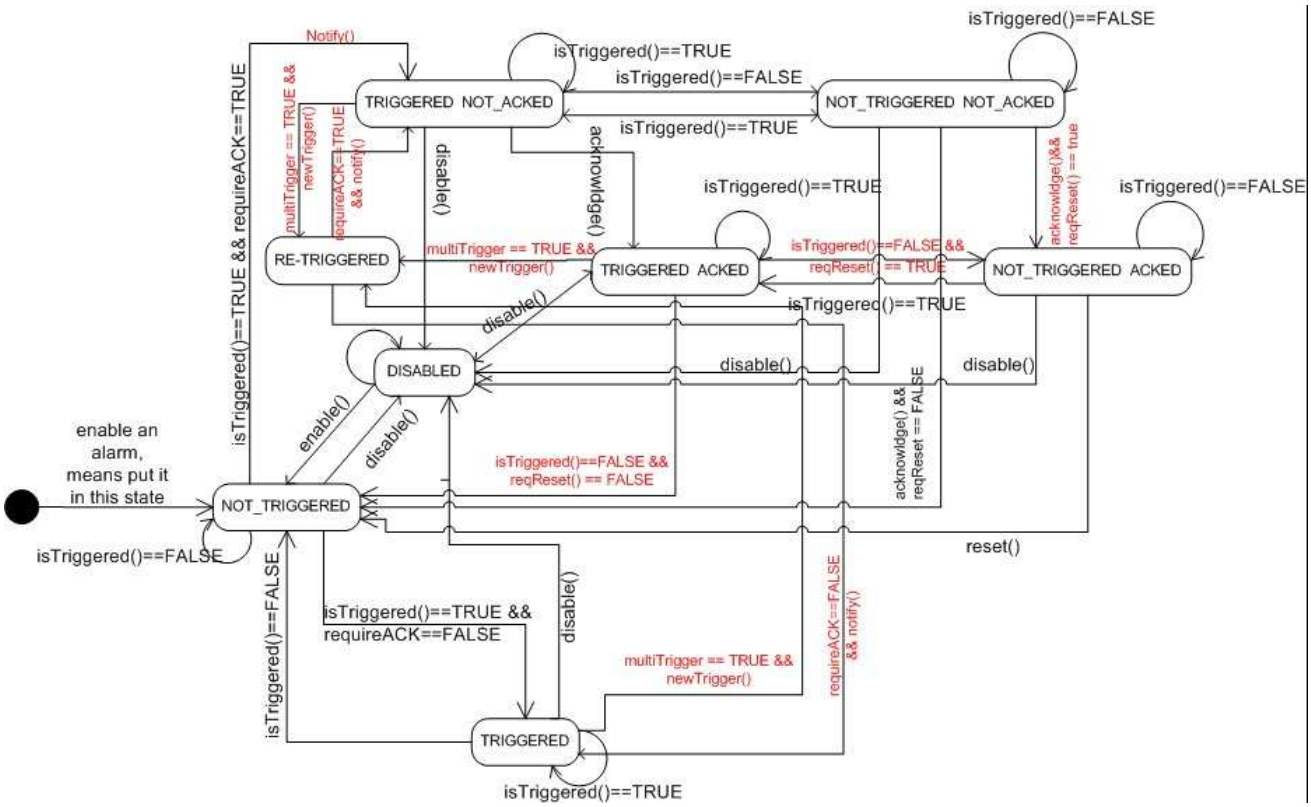


Figure 132

14.3 Setting Events

This chapter describes how to set Events in the Alarm Configuration Editor.

14.3.1 Log Events

Select the “Log" tab in the dialog box (as shown in figure). The list below this represents a set of conditions for which you may want to store the specific event in the Alarm History Buffer. Click the check boxes corresponding to the application requirements.

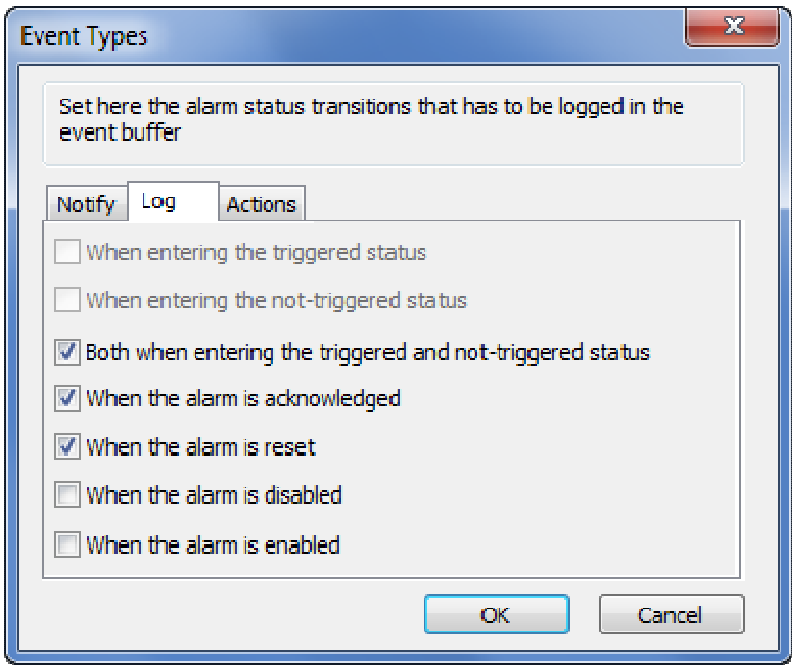

Event Types
Set here the alarm status transitions that has to be logged in the event buffer
Notify
Log
Actions
When entering the triggered status
When entering the not-triggered status
Both when entering the triggered and not-triggered status
When the alarm is acknowledged
When the alarm is reset
When the alarm is disabled
When the alarm is enabled
OK
Cancel

Figure 133

The Alarm Events History can be accessed by logging in a dedicated buffer called “Event Buffer"; to configure the Event Buffer, you have to double -click on “Buffers" in the Configuration Editor (as shown in the figure below). Here there is an Option for Selecting the storage type.

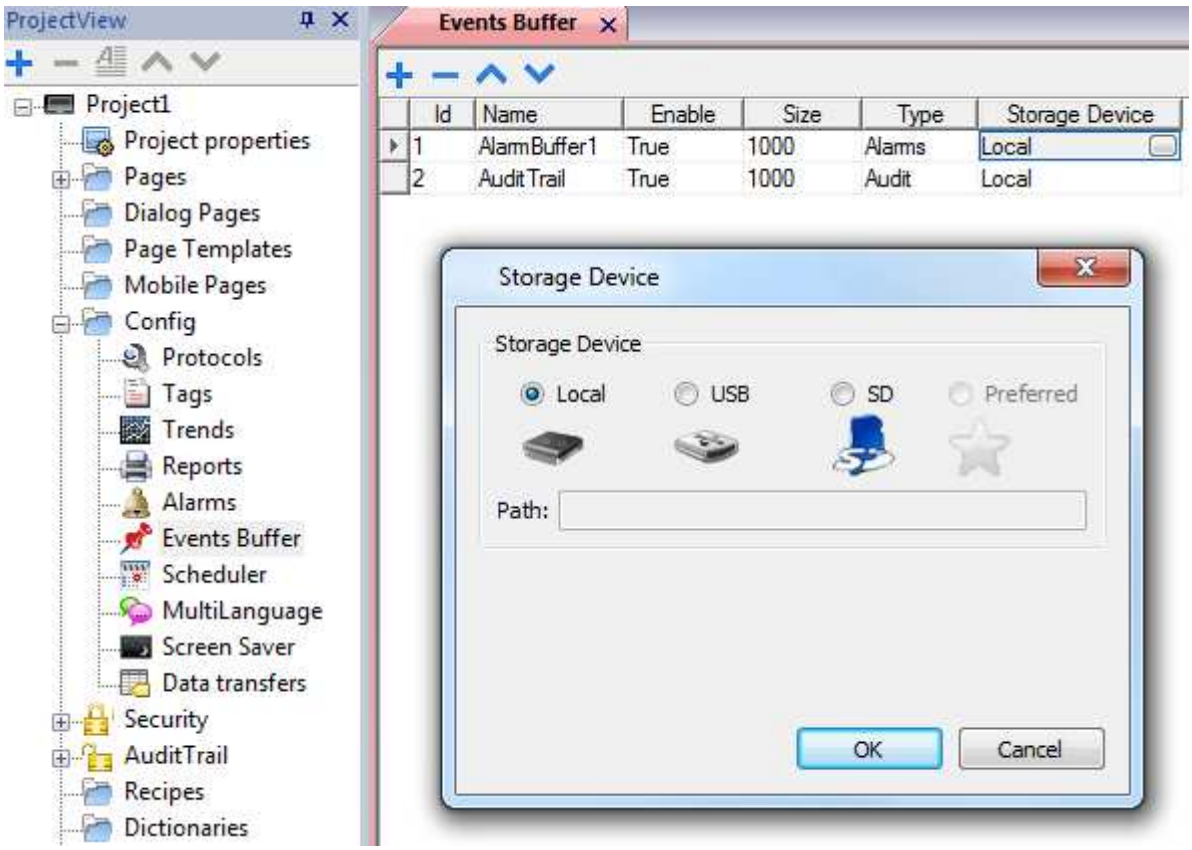


Figure 134

When the events' buffers are stored in persistent storages (Local, USB, SD etc), the system saves the file on disk every 5 minutes. However, events of type alarms are saved immediately.

14.3.2 Notify

The user can choose the conditions under which the Alarms should be posted in the Alarm Widget. This specifically refers to the default Alarm Widget, available in the Widget gallery. The user can decide when the Widget will be updated with a change of an Alarm Status. We recommend leaving the default settings here, and changing only those necessary for specific application requirements.

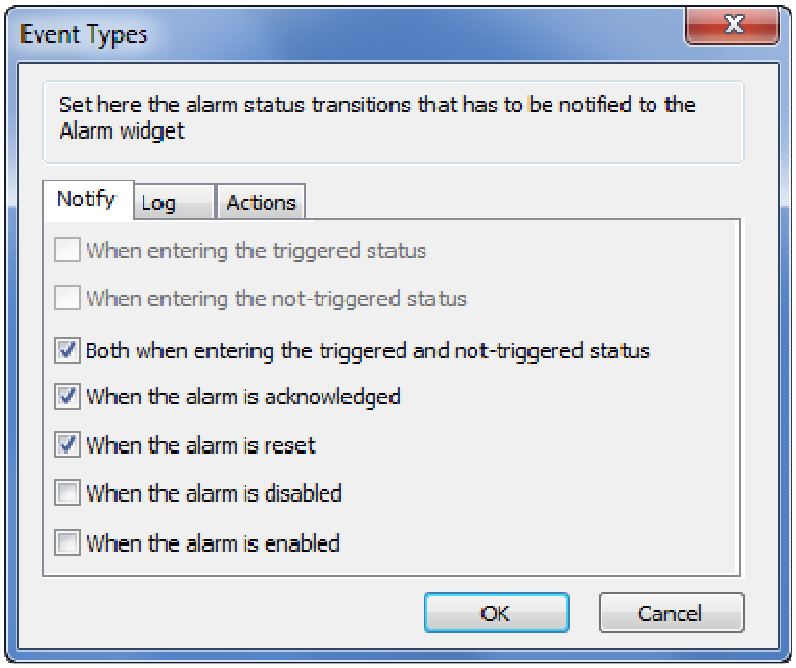

Event Types
Set here the alarm status transitions that has to be notified to the Alarm widget
Notify
Log
Actions
When entering the triggered status
When entering the not-triggered status
Both when entering the triggered and not-triggered status
When the alarm is acknowledged
When the alarm is reset
When the alarm is disabled
When the alarm is enabled
OK
Cancel

Figure 135

14.3.3 Actions

The user can specify the conditions under which the action(s), configured for the specific Alarm, must be executed.

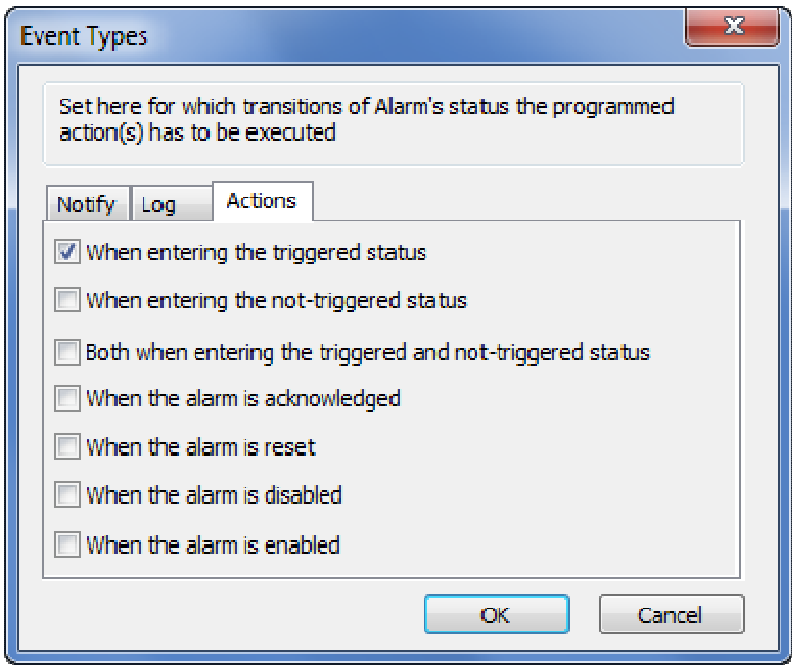

Event Types
Set here for which transitions of Alarm's status the programmed action(s) has to be executed
Notify
Log
Actions
When entering the triggered status
When entering the not-triggered status
Both when entering the triggered and not-triggered status
When the alarm is acknowledged
When the alarm is reset
When the alarm is disabled
When the alarm is enabled
OK
Cancel

Figure 136

By default, the actions are executed only when the Alarm enters the triggering condition; you may change this by configuring the system to execute the configured action also for the other alarm states available.

14.4 Active Alarms Widget

You can insert the **Active Alarms** Widget in a page to see the status of alarms and to acknowledge or reset or enable/disable alarms.

Figure 137

The Alarm Widget will display the Alarms in Runtime.

14.4.1 Filters

A Filter is available to show/hide just a subset of all configured alarms. Using the Combo box **Filter** it's possible for example to **Hide Not Triggered** alarms.

Another Filter (**Filter 2**) is available in widget properties and can be used to add a second filter based on another alarm field like Alarm name, or based on Severity or Description of an alarm.

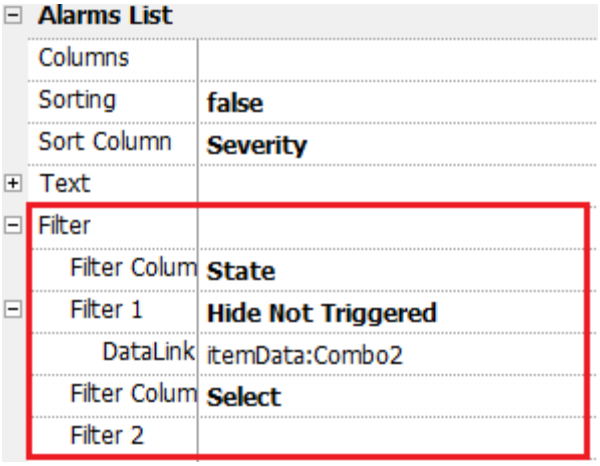


Figure 138

To customize a filter proceed as follow:

1) Select *Active Alarms* widget
2) Into the property pane select one of the two filters availables:
   Filter -> (*Filter Column 1* | *Filter Column 2*)
   and related value to filter (*Name* | *State* | *Value* | *Time* | *Description* | *Severity* | *Enable*)
3) Attach Combobox widget to *Filter 1* | *Filter 2 DataLink*.
4) Select ComboBox using "*Shift*" + "*Mouse Left Button*"
5) From property pane select *List* property and open dialog to customize combobox values
6) Using combobox configuration dialog, specify String List (options shown to user into the combobox) and regular expression to use for filtering values.

   More details about Regular expressions avialble here:
   http://www.gnu.org/software/gawk/manual/gawk.html

Follow examples of filters*:*

Filter By	**String List**	**Data List**
State	Hide Not Triggered	^((Not Triggered Acked|Not Triggered Not Acked|Triggered).*$)
Value	10 < Value < 20	^(1[0-9]$)
Value	20 <= Value <100	^([2-9].$)
Value	100 < Value < 200	^(1[0-9][0-9]$)
Value	Value 2?/3?/4?/5?	^([2-9].*$)

Value	Value >= 100	^([1-9][0-9][0-9].*$)
Value	Value >= 20	^([2-9].*$\|[1-9][0-9][0-9].*$)

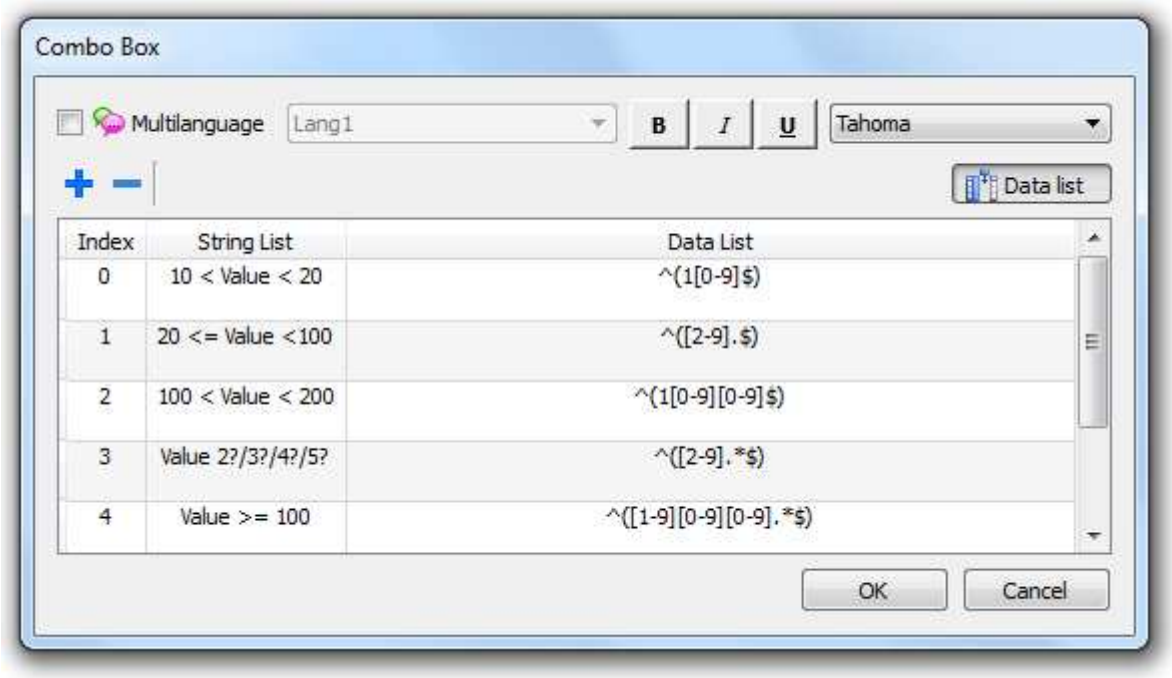


Figure 139

14.4.2 Sorting

You can enable or disable the column sorting option, available at Runtime for the Alarms Widget, by clicking on the column header. The sorting order is based on the string sorting.

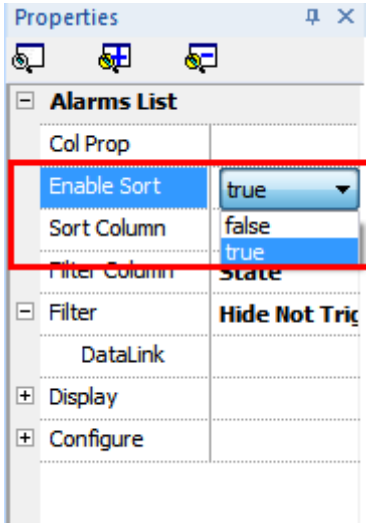


Figure 140

NOTE *Starting from version 1.80, the Alarms' widget provided in the gallery no longer has the "Priority" column. The widget has a new column called "Severity" which comes by default next to the ID column. Severity column takes the values from the Severity settings from the Alarm Editor.*

14.5 Alarms History Widget

JMobile Studio automatically logs the Alarm list based on the Flag Settings set in the Alarms Editor, under "Log Event Types". To see the Historical Alarm list, you can use **Alarms History** Widget.

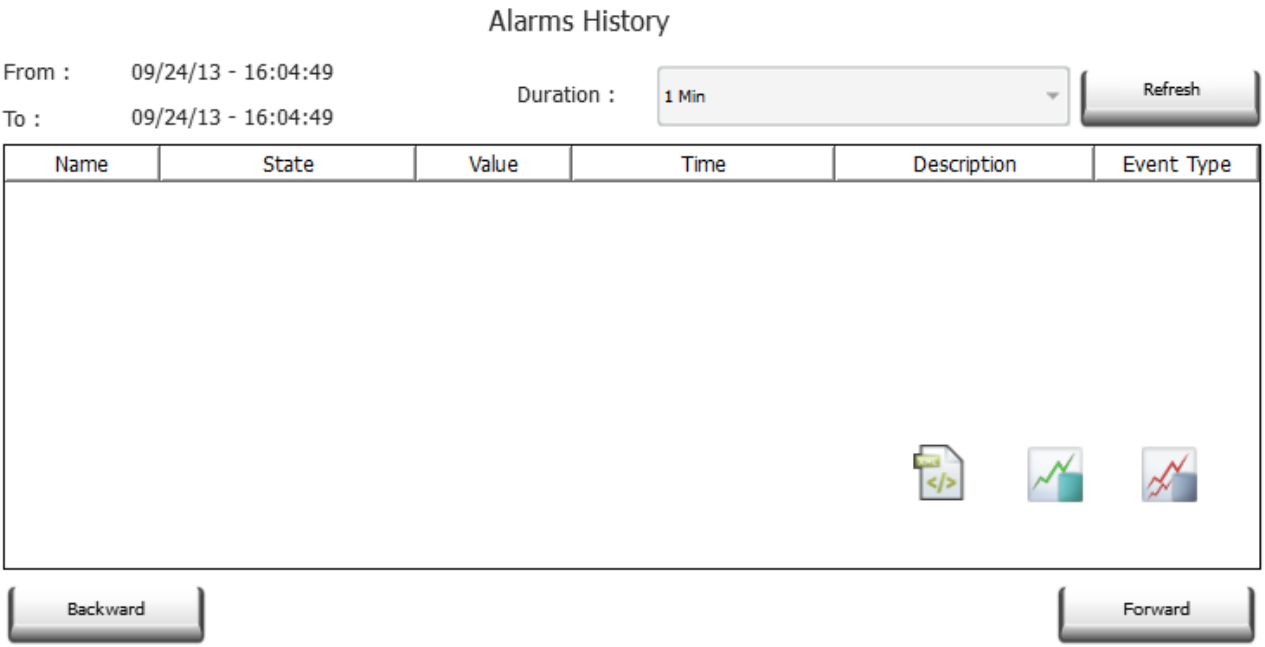


Figure 141

The selection of the Event Buffer is available in the property panel (as shown in the figure).

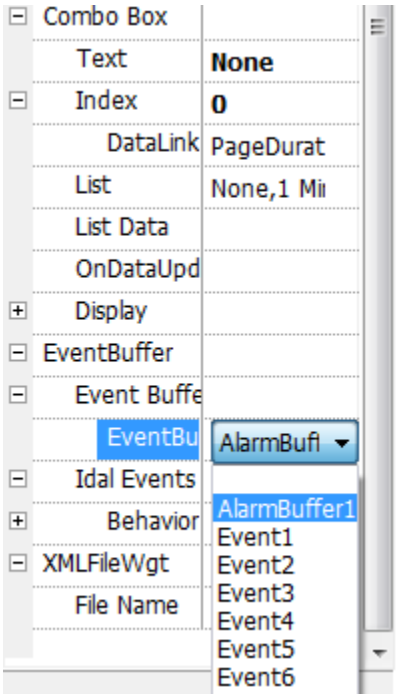


Figure 142

NOTE *For each of the different Alarm Buffers, a specific Event Widget must be configured for the project; the current version of the Event List Widget does not allow you to switch between buffers.*

14.6 Managing alarms at Runtime.

When an Alarm is triggered, the Alarm will be displayed in the Active Alarms Widget. The Widget allows you to acknowledge and reset the Alarm.
The Alarm display can be filtered by “Hide Not Triggered", “Show All" and other custom filters.
Please note that the visualization of the Alarm Widget is not automatic. If the Widget has been placed on a certain page, when an alarm is active, you must add a dedicated action that will go to the page showing the Alarm widget.

14.7 Enable/Disable Alarms at Runtime

You can enable or disable the alarms at runtime. If you want to disable an alarm, just uncheck the alarm from the Enable column in the Alarm Widget and execute the Save command. This way the alarm will not get triggered and the disabled alarm will not be displayed at Runtime.

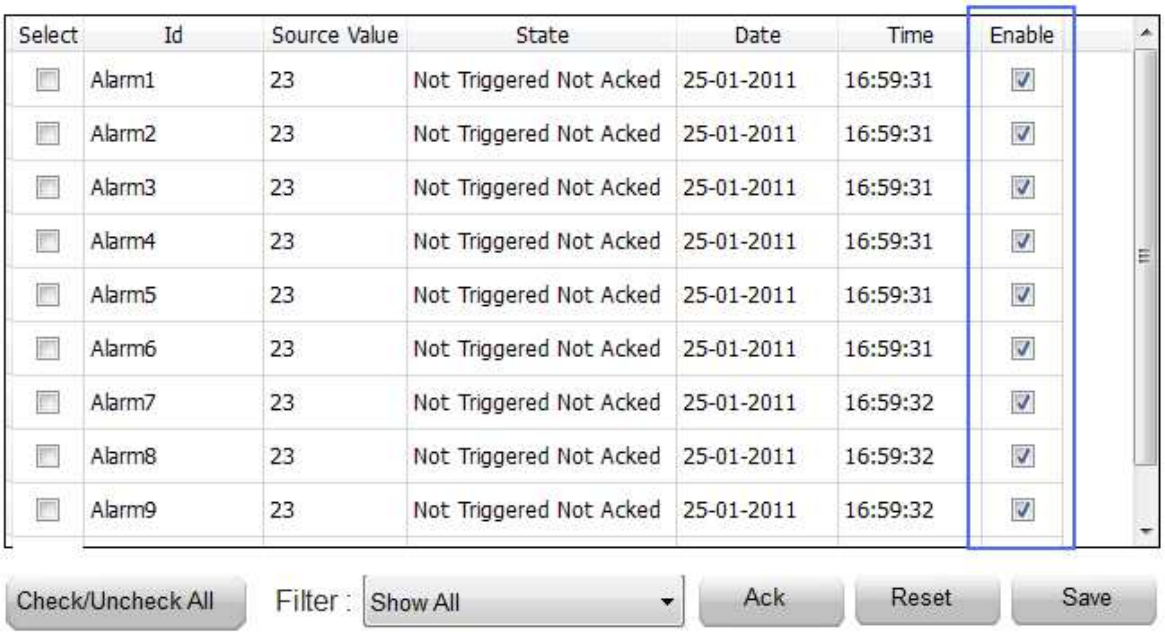

Select
Id
Source Value
State
Date
Time
Enable
Alarm1 23 Not Triggered Not Acked 25-01-2011 16:59:31
Alarm2 23 Not Triggered Not Acked 25-01-2011 16:59:31
Alarm3 23 Not Triggered Not Acked 25-01-2011 16:59:31
Alarm4 23 Not Triggered Not Acked 25-01-2011 16:59:31
Alarm5 23 Not Triggered Not Acked 25-01-2011 16:59:31
Alarm6 23 Not Triggered Not Acked 25-01-2011 16:59:31
Alarm7 23 Not Triggered Not Acked 25-01-2011 16:59:32
Alarm8 23 Not Triggered Not Acked 25-01-2011 16:59:32
Alarm9 23 Not Triggered Not Acked 25-01-2011 16:59:32
Check/Uncheck All
Filter :
Show All
Ack
Reset
Save

Figure 143

Later, if you want to enable again the Alarm, select the Alarm and check the Enable check box. Then execute the Save command. The Alarm will now be subscribed and subject to being triggered.

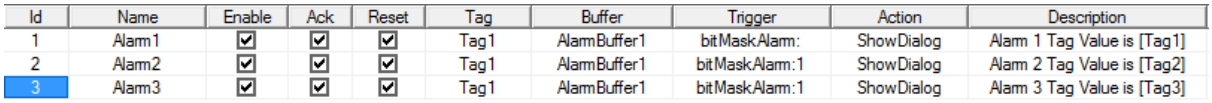
14.8 Live Data in Alarms Widget

This feature is used to view the live Tag data value inside the alarm description. It is applicable for both Active Alarms and History Alarms widget.

To configure the live data visualization in the Alarm Widget, follow a simple syntax rule.
The Tags to be included must be specified in the alarm description string, including the Tag names in square brackets:

[Tag name]

An example is shown below.



Id	Name	Enable	Ack	Reset	Tag	Buffer	Trigger	Action	Description
1	Alarm1	☑	☑	☑	Tag1	AlarmBuffer1	bitMaskAlarm:	ShowDialog	Alarm 1 Tag Value is [Tag1]
2	Alarm2	☑	☑	☑	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 2 Tag Value is [Tag2]
3	Alarm3	☑	☑	☑	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 3 Tag Value is [Tag3]

Figure 144

During Runtime in the Alarm Widget, the markers and Tag name will be replaced in the description column by the actual value of the Tag. The Widget automatically refreshes and shows the current values of the Tags in the Widget.
In History Alarm Widget, it will show the value of the Tag at the moment the alarm was triggered.
Into the CSV file resulting from the Dump of the alarms events list, the Tag values can be seen in the description column.
Result will be displayed as shown in the figure below.

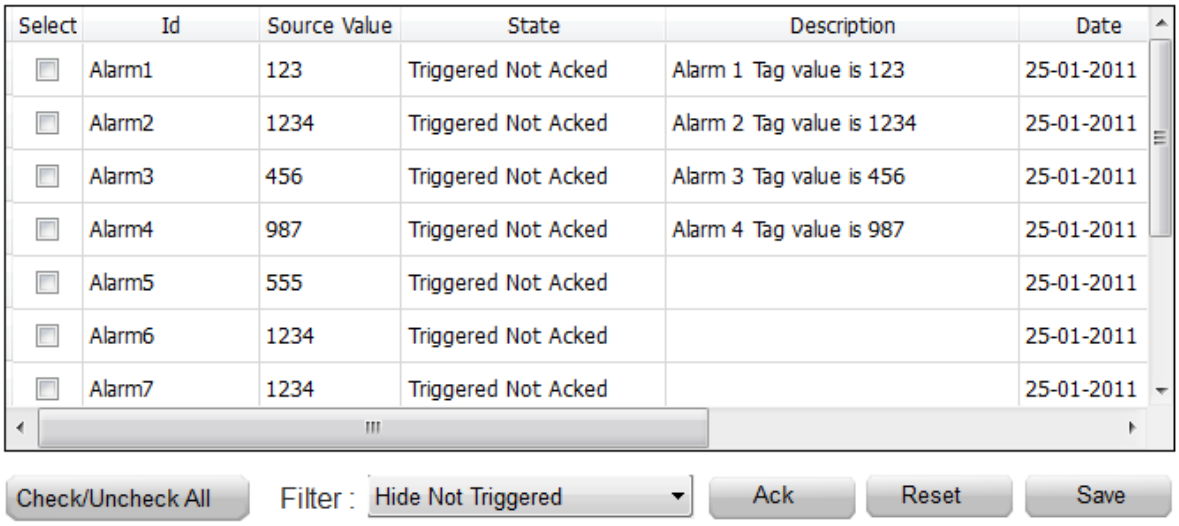


Figure 145

NOTE *The ability to store the alarm description with tag values in the event buffer is a feature supported starting from version 1.80*

NOTE *use '\' before '[ ]' where there is a need to show the '[ ]' in the description string. So, if the string to show is [Tag[1]], the correct syntax to use is [Tag\[1\]]*

14.9 Exporting Alarm Buffers as CSV file

The historical alarm list (the event buffer) can be exported using the action called “DumpEventArchive”...

NOTE *The tag values included in the Alarms description are also included in the event log stored in the event buffer The tags are sampled at the moment the alarm is triggered and that is the value recorded and included in the description. In the Alarm description, displayed by the Alarm widget, the value may change because it is constantly updated, but no additional values are recorded. This feature is supported starting from version V1.80.*

15 Recipes

Recipes are a feature for organizing data storage in the HMI device and include services for exchanging data with connected controller devices.
This data can be written to the controller, and, conversely, the data can be read from the controller and saved back on the HMI panel. This concept offers you a powerful way to extend the capabilities of the controller. This is especially true for controllers that have a limited amount of memory.

The Recipe memory is the physical storage for the Recipes. The “Recipe Tag" block basically identifies the “current Recipe". From the Recipe memory, you select one Recipe data record or Recipe set and designate it “current/active Recipe". Then, you can transfer this recipe data, to or from the controller. Recipe tags can be displayed and edited on a page.

Currently, the Recipe data is configured in the JMobile Studio workspace and the user can specify default values for each element of the data records. On Runtime, the data can be edited; this new data is saved to a new and separate data file, different from the original one containing the default values. Any change to recipe data is stored on disk. The use of a separate data file on Runtime ensures that modified Recipe values are retained throughout different project updates. In other words, a subsequent project update does not influence the Recipe data modified by the user on Runtime.

NOTE *To reset the recipe data to the default values, there is a dedicated action called “Reset recipes”; see below in this chapter for further information.*

The User can also select where the Recipe needs to be stored. There are three options for this: FLASH, USB, and SD Card. The user can select any one.

You can configure Recipes by adding the required controller data items to a page from the Recipe Widget. A Recipe can be associated with a particular page and is composed of all the Recipe data items on that page. Recipe data items contain all the information associated with normal controller data items; but, rather than the data being read and written directly to the controller during the course of normal operation, the data is instead read from and written to the panel memory that is reserved for the data item.

This chapter describes how to configure and use the Recipes in the JMobile Studio application.

15.1 Recipe Configuration Editor

In the Project View pane, select Recipes and right click. Then choose Insert Recipe if you want to create a new Recipe. The newly added Recipe item will be added in the project workspace.

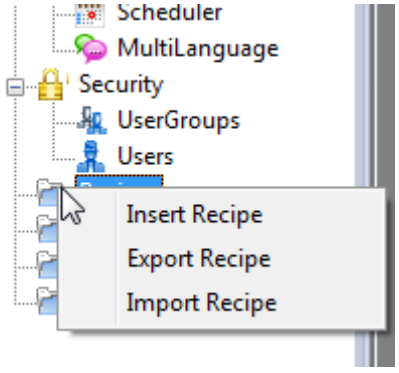


Figure 146

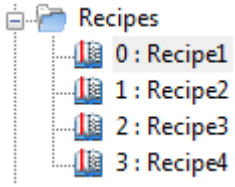


Figure 147

Double click on the Recipes to open the Recipe Editor, as shown in the figure below.
Add the recipe elements by clicking the “+" button, and then link the tags to the recipe element.

By clicking on the button “Storage Type” you can select where to store recipe data.

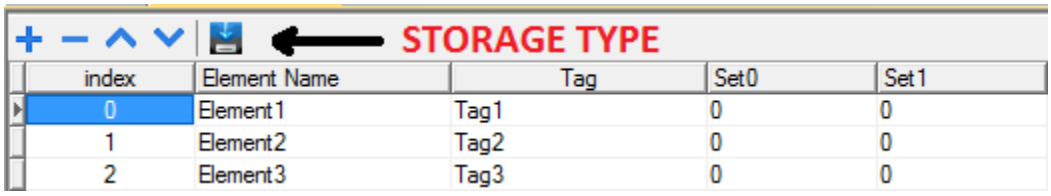


Figure 148

A dialog in which the selection can be made will open. See the figure below.
For USB and SD card you can provide the folder location.

WARNING *Recipes configuration files are created automatically when the project is saved. Recipes files are saved into the subfolder **data** of the project folder into the PC by JMobile Studio. When external storages are used, please copy this folder into the external storage selected. Default path is “/Storage Card/data” for SD or “/USBMemory/data” for USB storage. However, a subfolder of it can be used like “/USBMemory/MyRecipes/data”.* The subfolder name “data” cannot be changed and is required for the recipes to work.

Figure 149

15.2 Configuring Recipe Sets on the Page

The number of parameter sets can be changed in the “Number of sets” field in the property pane. From there you can also change the name of each Recipe set.
Recipe values for all the parameter sets can be entered into the Recipe Editor window.

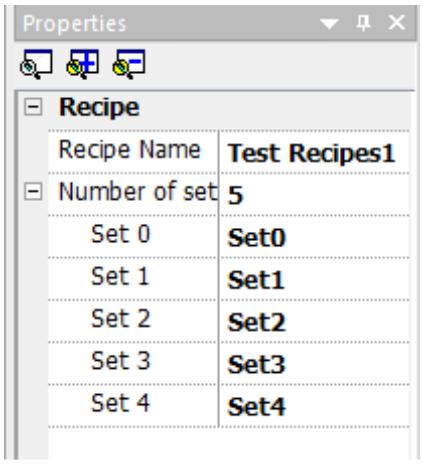


Figure 150

15.3 Defining Recipe Fields

The user can define the Recipe field on the page by using the numeric field Widget from the gallery and attach the Tags from the Recipe data source. The figure below shows an example of a Tag attached to a Recipe field.

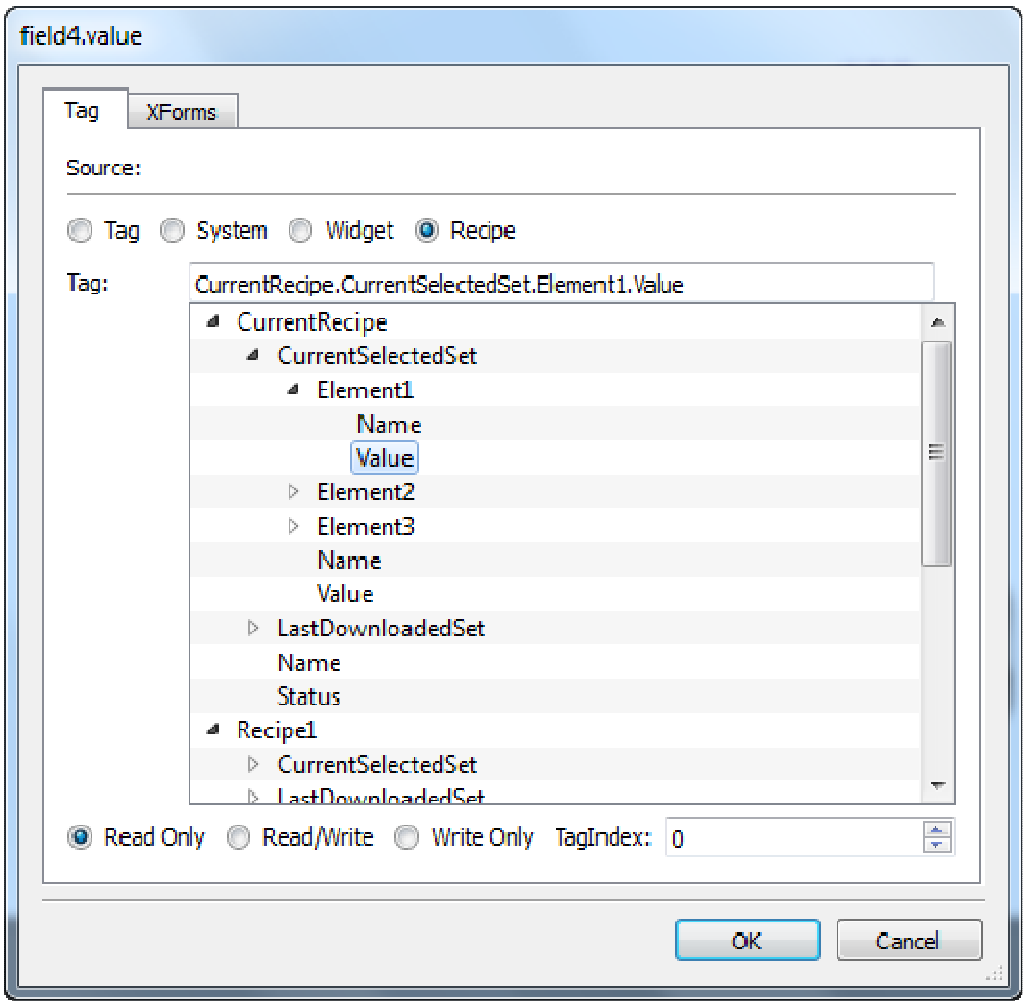


Figure 151

The “Attach to” Dialog allows you to attach to the numeric field all the different Recipe variables, such as:

- Current Recipe ->Current selected Recipe set-> Element -> value (or) name
- Selected Recipe -> Selected Set0 -> Element -> Value (or) Name
- Selected Recipe list
- Currently selected Recipe list
- Recipe Status

When the numeric fields are defined as Read/Write, the default Recipe data can be edited at Runtime. As explained in the introduction, these new values are stored in a separate file as modified Recipe data.

15.4 Recipe Status

After every *Recipe Upload* or *Download*, or *Recipe set modification*, the **Recipe Status** parameters contain a value with the result of the operation.
The following are the values and conditions for the **Recipe Status** system variable.

Code	Function	Description

0	Set modified	Current selected set changed.
1	Download triggered	Triggered a download request.
2	Download Done	Download action completed.
3	Download Error	Error occurred when doing download - errors like unknown set, unknown recipe, controller not ready, Tags write failed etc.
4	Upload triggered	Triggered an upload request.
5	Upload done	Upload action completed.
6	Upload Error	Error occurred when doing upload - errors similar to download errors.
7	General Error	Errors like data not available.

NOTE *When the panel starts up the value of **Recipe Status** is **0**.*

15.5 Configuring Recipe Widget for Runtime Execution

Two default Recipe Widgets are available in the advanced Widget Gallery category.
The “Recipe Set" Widget allows you to select a Recipe set for the upload and download operations. If you have more than one Recipe in the project, then the “Recipe Menu" Widget can be directly used to manage all the Recipes from a single Widget, listing Recipes and selecting the sets for each Recipe.

Figure 152

15.6 Configure Recipe Transfer Macros.

The Recipe transfer action can be completed through the action list dialog. The transfer of Recipes can be achieved by any of the following methods:

Attaching an action to an event for button or switches

Configuring the action from the Alarms action list.

Using the Scheduler actions list.

Description of actions available for Recipes is included in the relevant chapter.

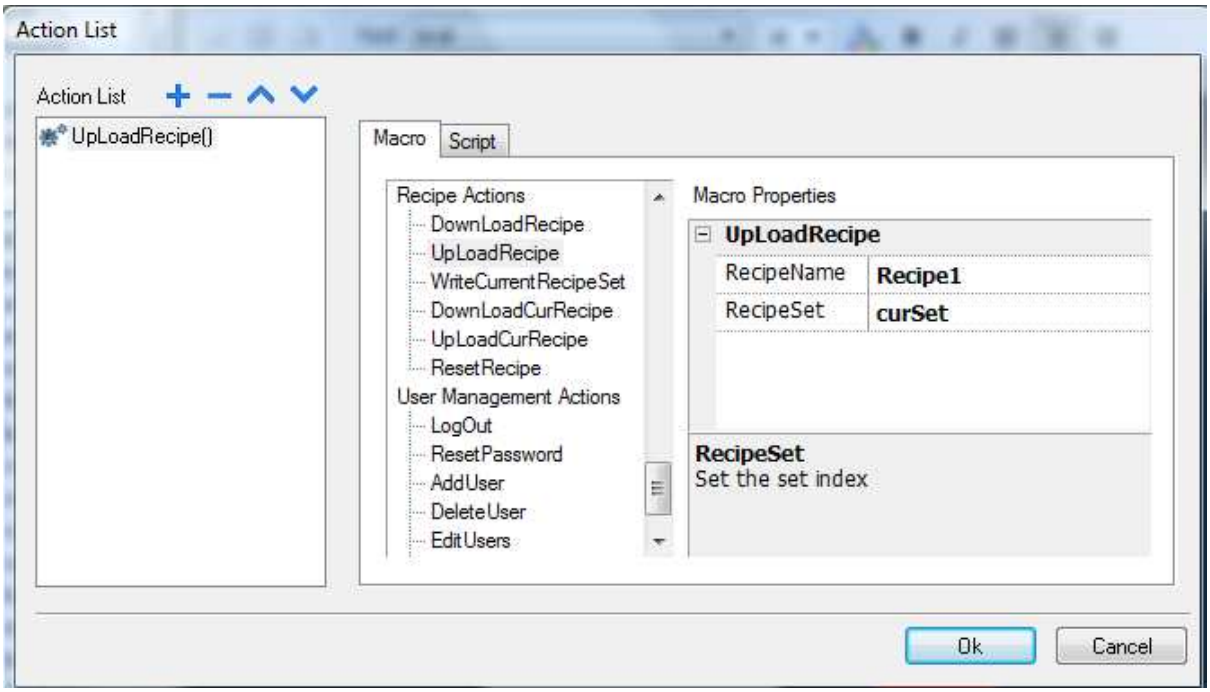


Figure 153

15.7 Upload or Download Recipes during Runtime

15.7.1 Recipe Download through Recipe Widget in Runtime

Drag and drop the Recipe Widget (as described in the Chapter" Configuring Recipe Widget for Runtime Execution") into the project to execute the Recipe transfer in Runtime. Select the Recipe from the drop down box, and select the Recipe set from the set dropdown list. Then press the "Download" button to download the current selected Recipe set, or press the "Upload" button to upload the current selected Recipe set.

15.7.2 Recipe Download or Upload through Recipe Transfer Macro in Runtime

The Recipes can be Downloaded or Uploaded through the Recipe transfer macro. At runtime, execute the macro (if the macro is programmed with a push button, then press the button). The Recipes data will then be transferred to the controller, or uploaded from the controller, depending on the action programmed. The figure below shows a simple example of a project using Recipes at Runtime.

Figure 154

15.7.3 Backup and Restore of Recipes Data

The Recipe data stored in the HMI device can be exported for backup purposes and later restored. Please refer to the actions “Dump Recipe Data“and “Restore Recipe Data” for further information.

16 Trends

Trending is a method of sampling and recording the values of a specified Tag according to sampling conditions (normally, the time).

Trending is divided into two main parts: Trend acquisition and Trend viewing. Trend acquisition (programmed with the Trend Editor) collects the data into a database. The Trend viewer (Trend Widget) displays the data from this database in a graphical format.

16.1 Real-Time Trend

In real-time Trend, the data will be presented directly in the Trend window, and the changes to the live data can be seen directly in the format of a curve on the Trend window. Users can manage the process by seeing the Trend on the HMI. The real-time Trend Widget is just a viewer for a Tag, and it does not refer to any saved data in any buffer. Any curve plotted is lost when the page containing the Widget is changed.

To configure the Real-time Trend, just drag and drop the Real-time Trend Widget from the gallery.

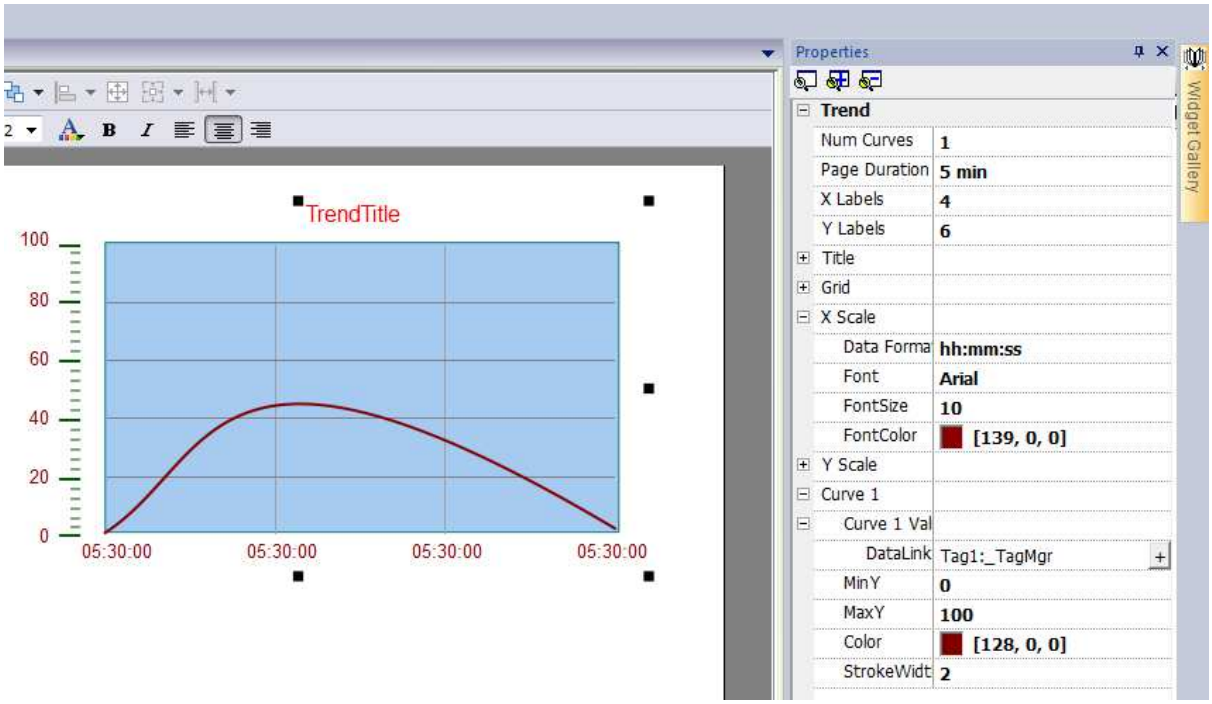


Figure 155

Select the Trend Widget and, in the properties pane, attach to the “Curve x Value" property the Tag for which you want the data to be plotted. Data is always plotted against time.

Following is the list of main parameters of Real-time Trend widget:

Num Curves Number of trend curves in the Trend window. A maximum of 5 curves can be configured in a Trend window.

Page Duration	Time range of the X-Axis. However, you can dynamically change the page duration in Runtime with the Date Time combo widgets, attaching it to the Trend window page duration properties.
X Labels	Number of Labels in the X-axis scale
Y Labels	Number of Labels in the Y-axis scale.
Title	Trend title and font properties (font size, label, etc.)
Curve x	Tag or ì Trend buffer that will be plotted into the trend window.

Scaling can be applied to the Tag values. To apply scaling, use the X Forms attached to dialog.
You can set the Minimum or Maximum of the curves. You can also attach a Tag to these minimum and maximum properties. This enhances the ability to change the min and max dynamically in the Runtime. Also you can modify the properties, such as colors, update time, number of samples, etc. of the Trend curves through the property view.

16.2 History Trend

If you want to analyze the data at a later time, the Trend data will need to be stored. For this purpose you use History Trend. When you select History Trend, you can store the data information with reference to time.

The first step in creating a History Trend is to create a Trend Buffer.
The purpose of the trend buffer is to save a sequence of values of a specified Tag in order to record the state of the tag while time changes. Once values are stored in the buffer, a dedicated widget, called History Trend viewer, can be used to display the curve in a graphical format.
The History Trend viewer is available in the widget gallery.

In the History Trend widget the start time of the Trend window will be the current time and stop time will be the current time + duration of the window.
The plot starts from the left end of the Trend window as in the figure below. The graph will be automatically refreshed during a certain interval of time, until the stop time. When the curve reaches the stop time, the graph will scroll left and the update of the curve will continue until it again reaches the right side of the viewer. At that moment a new scroll is automatically done and the process repeats.

NOTE *Automatic refresh is an option available starting from version 1.80.*

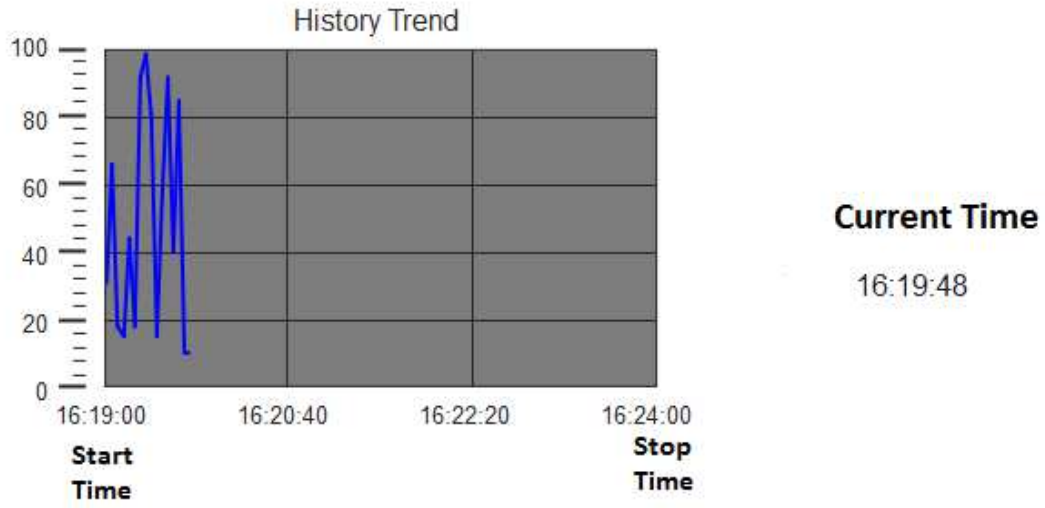


Figure 156

16.2.1 Trend Editor

Historical trends require a proper configuration of trend data buffer. Trends' buffers are configured using the Trend Editor.

Trend buffers are stored in data files. There is an option to store these files on the internal storage (Local), USB Memory, SD card or custom folders based on target platform.

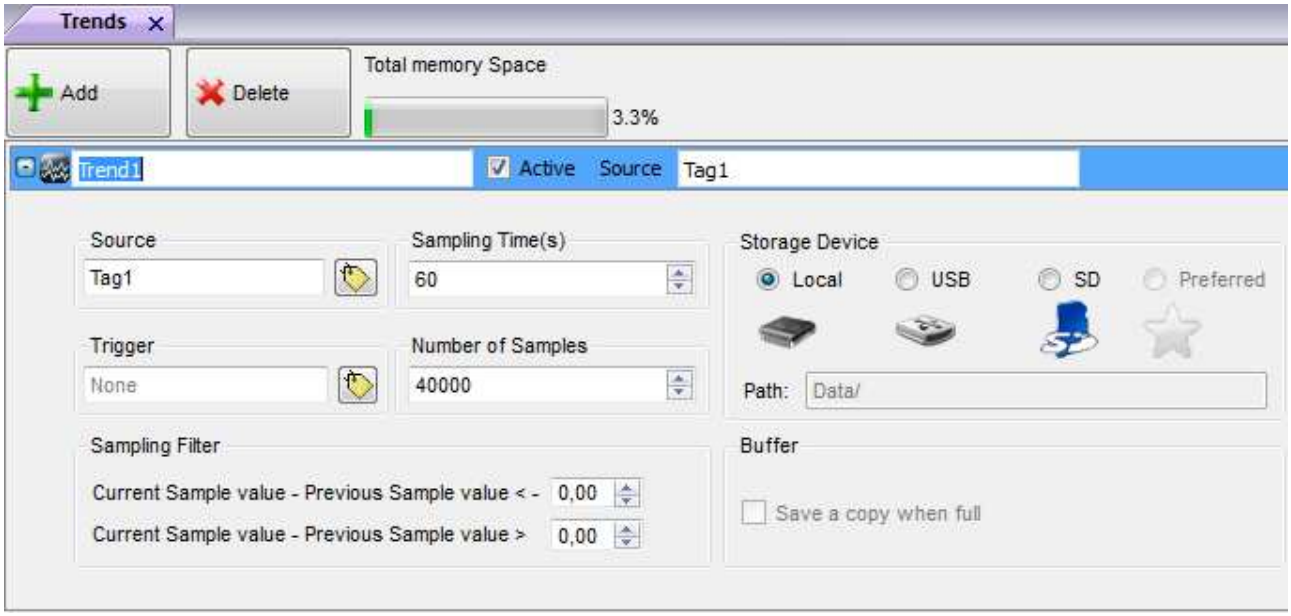


Figure 157

In the Project View pane, double click Trends to open the Trend Editor. Then add the trend buffer, by selecting the "+ Add" button on the editor. By clicking "+" near each trend buffer, the corresponding buffer configuration is expanded.

The “Total Memory Space” bar shows how much memory has been used by the trend buffers currently configured. The max number of samples allowed for a project is 1200000. The memory use is the percentage of this number. As in the Figure above, suppose the total number of samples used in the project is 80000. Then the total memory used will be shown as 3%. This is calculated by the formula

$$\text{Total Memory Space} = \frac{\text{Total Number of Samples used in the Project}}{\text{Max Number of Samples allowed for a Project}} * 100$$

As we Increase the number of samples, the percentage of usage also increases and this will be shown in the bar.

The following are the properties of each trend buffer in the Trend Editor:

Trend Name	Defines the trend buffer name, which will appear when you define the buffer to a trend window property pane. A default name is assigned by the system; the name can be modified by the user.
Active	Specifies if the trend runs by default when the system starts up. NOTE *The trend buffers cannot be activated during Runtime*
Source	This combo list allows selecting the Tag which is sampled by the Trend manager system.
Sampling Time	Samples are collected and stored in the disk data file on a cyclical basis. Default sampling condition is the time; the sampling time specifies the sampling period in seconds.
Trigger	When the Trigger tag is specified, the source tag is not sampled on a cyclical basis but on the Trigger tag value change. In any case, the samples are plotted with respect to the time. The Trigger tag and source tag can be the same.
Number of Samples	This represents the buffer size expressed in samples.
Storage Device	This is an option to select where the trend buffer data file will be stored
Buffer	Trend data is organized as a FIFO queue. Once the buffer gets full, the oldest values will be erased to create space for storing the new values. If **Save a copy when full** is selected, when the buffer gets full, before overwrite it, system create a backup copy of it into external storages.
Sampling Filter / Trigger Filter	When the triggering condition is the time, a new sample is considered significant (and then stored) only if its value, in comparison with the last saved value, goes out from the specified boundaries. In case the triggering condition is based on a trigger tag value change, the boundaries are applied to the trigger tag value.

16.2.2 Configuring Trend Window for History Trends

The History Trend widget (trend window) is the area used to display the trend buffer in a curve format. After configuring the trend buffer in Trend Editor, you can use the Historical Trend viewer widget to plot the trend curve on the screen. From the trend gallery page, drag and drop the “History Trend" widget to the page.

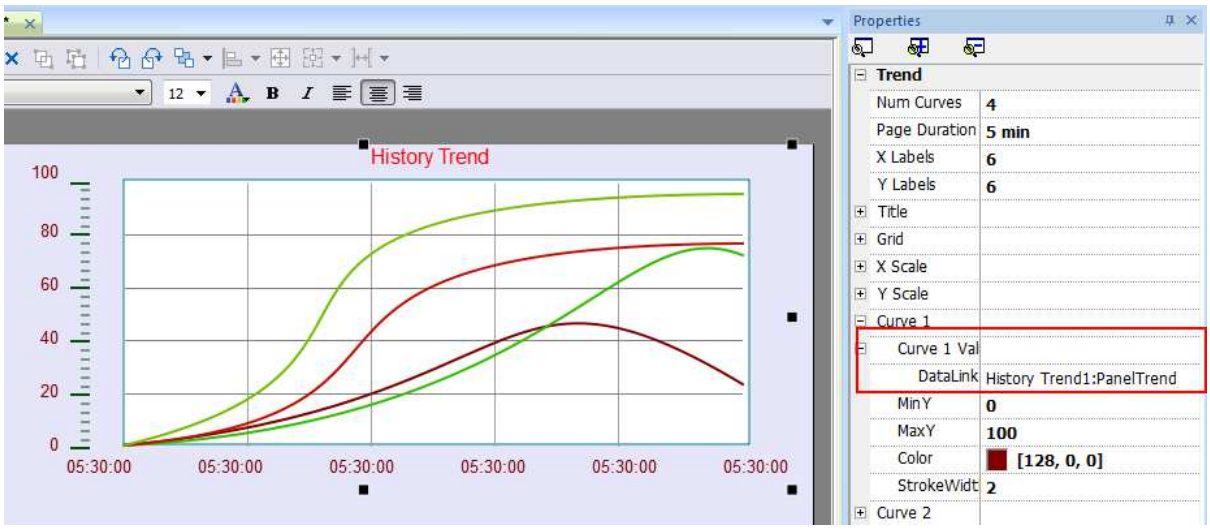


Figure 158

Then, in the property pane of the Trend window, attach the trend buffer to be plotted in the trend window (as shown in the figure below).

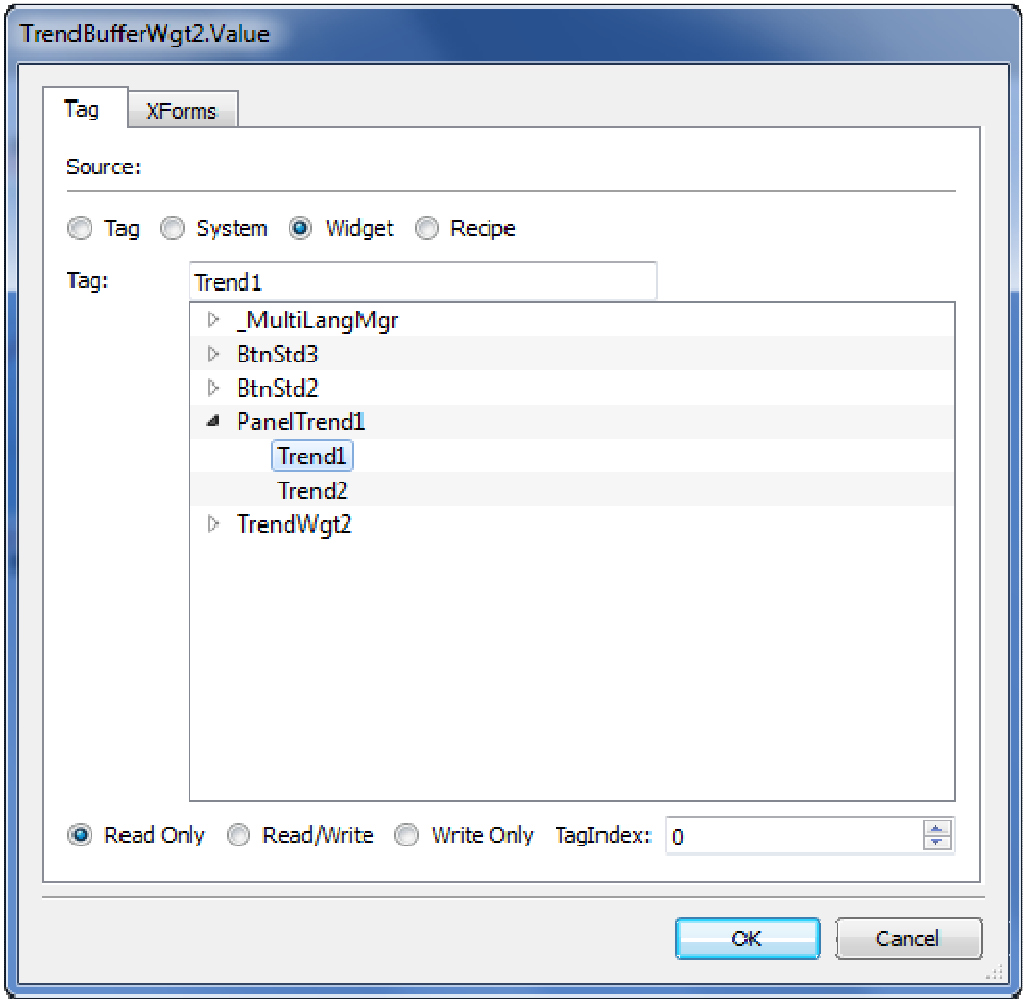


Figure 159

16.3 Trend Window Properties

With the help of the property pane of the trend window, you can customize the Trend window properties, such as, X Axis time, Y Axis value, number of trend curves, changes to the labels, grids, number of samples, etc.

16.3.1 Request Samples (Advanced View)

In the “Curve x” category there is one property called “Request Samples” as shown in the figure below.

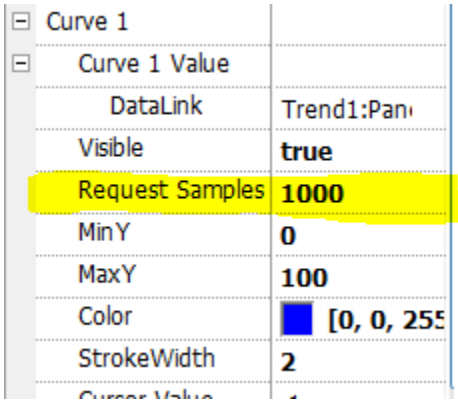


Figure 160

This property represents the maximum numbers of samples read by the widget at one time from the buffer data file; this block size can be adjusted to fine tune performances in trend viewer refresh, especially when working with remote clients. The default value is normally a good compromise for most cases.

16.3.2 Color Bands

Color Bands property of trends allow users to color the graph background based on the day of the week (Sunday..Saturay) and time (0..23).

Using the button “+” , you can be add as many color bands (colors) as needed.
Select multiple cells and click on a color band to assign color to that range of time.

NOTE *Color Bands feature is working just using Local Time in trend viewer (not Global).*

Figure 161

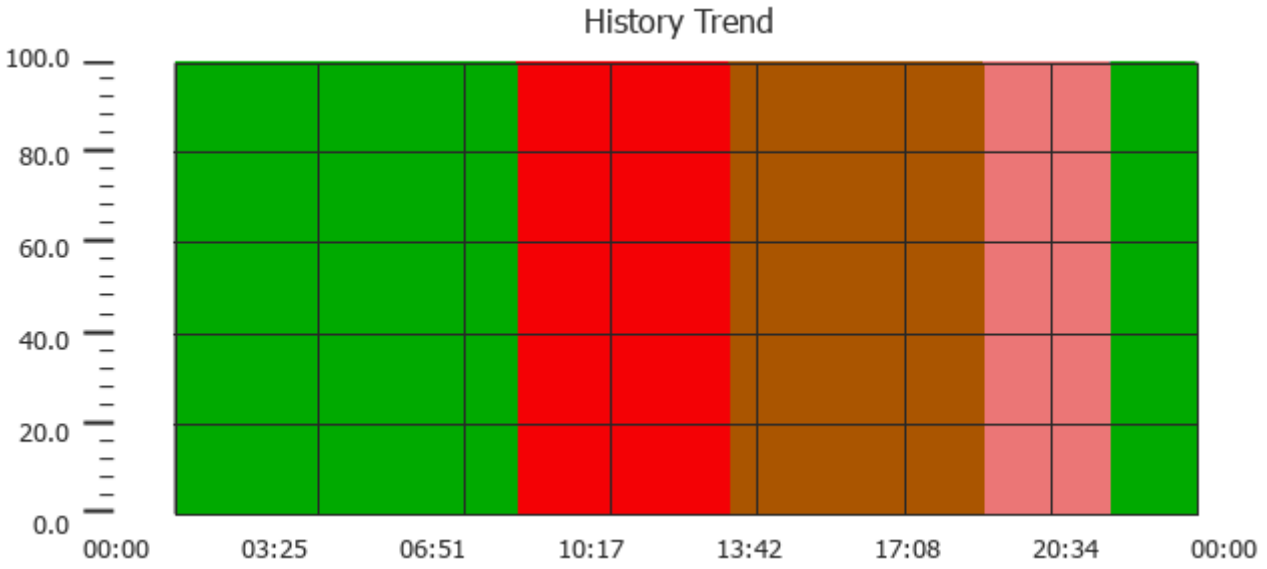


Figure 162

16.4 Trend Cursor

The Trend Cursor allows you to see the trend value at a point. Use Show Trend Cursor macro and Scroll Trend Cursor macro to enable the Trend cursor and move it to the required point to get the value of the Curve at particular instant in time.

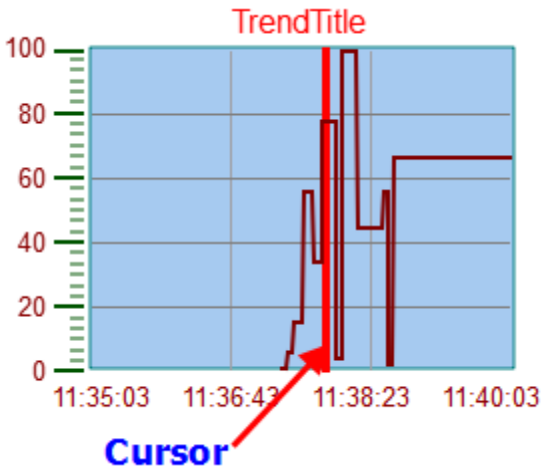


Figure 163

To display the value of the Trend Cursor on the page, define a numeric field and attach the Cursor Value Widget Tag (as shown in the figure below). This is the Y axis Value of the Cursor.

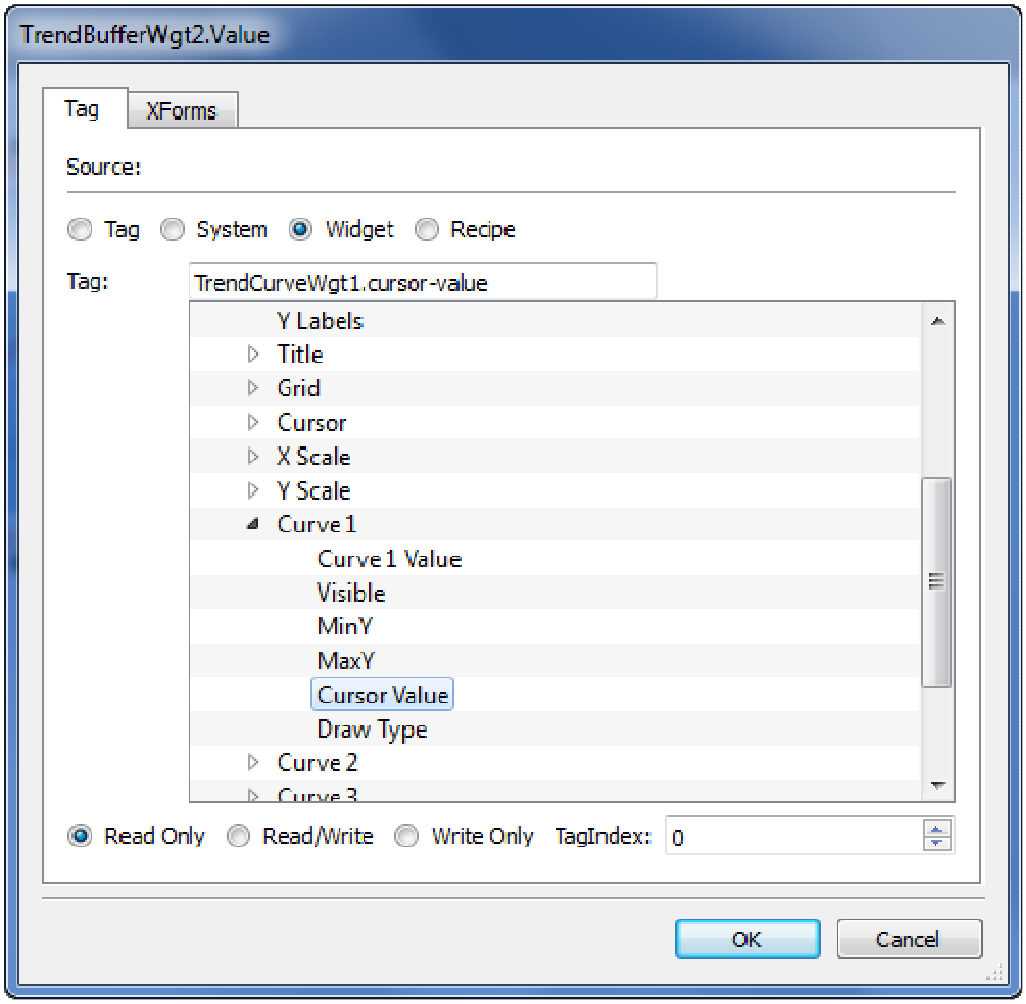


Figure 164

To get the Time at the Particular Point where the Cursor is placed, define a numeric field and attach to the “Widget Tag” as shown in the figure below.

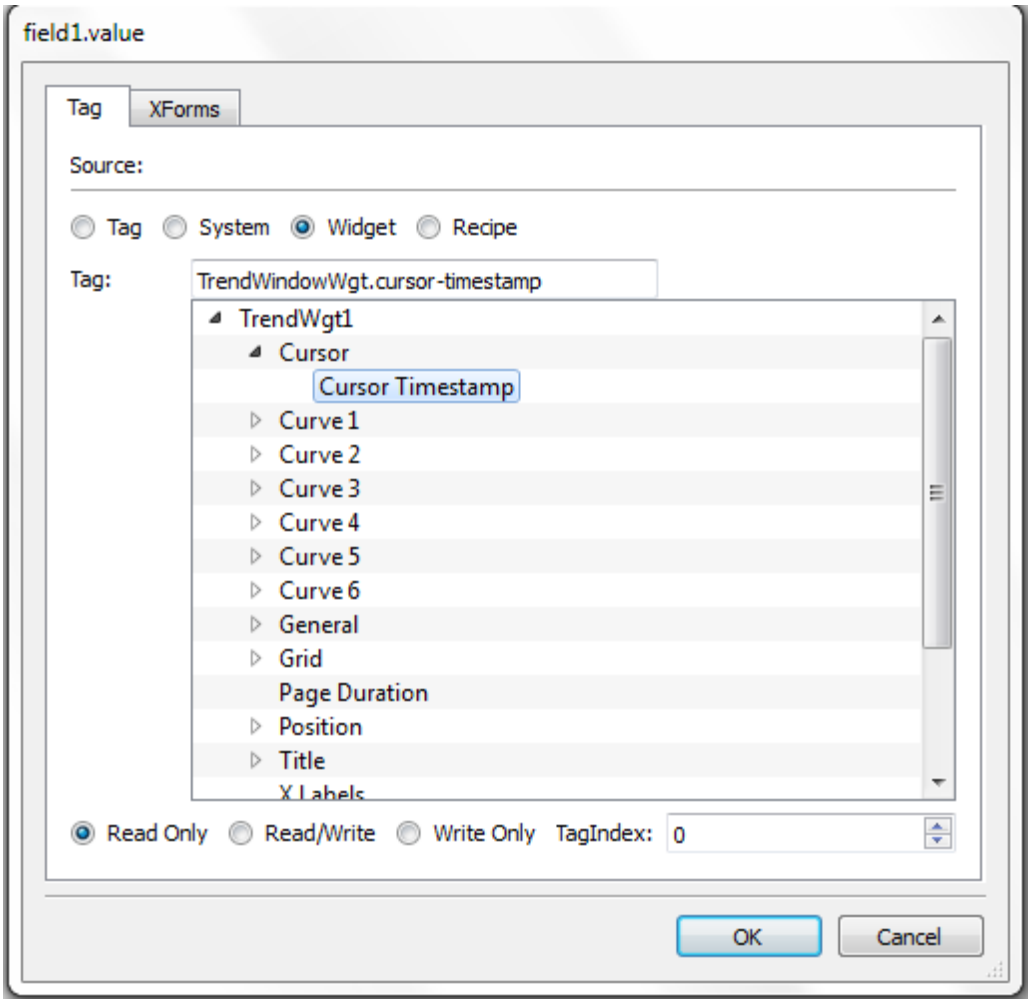


Figure 165

The Widget tag shown in the figure above represents the X axis cursor value for the trend window.

16.5 Exporting Trend Buffer Data to CSV file

The trend buffers stored in the selected media can be exported to CSV file using dedicated actions. Please refer to “Dump Trend” for further information.

17 Scatter Diagram / XY Graph

A scatter diagram is a type of mathematical diagram using Cartesian coordinates to display values for two variables from a set of data. The data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.

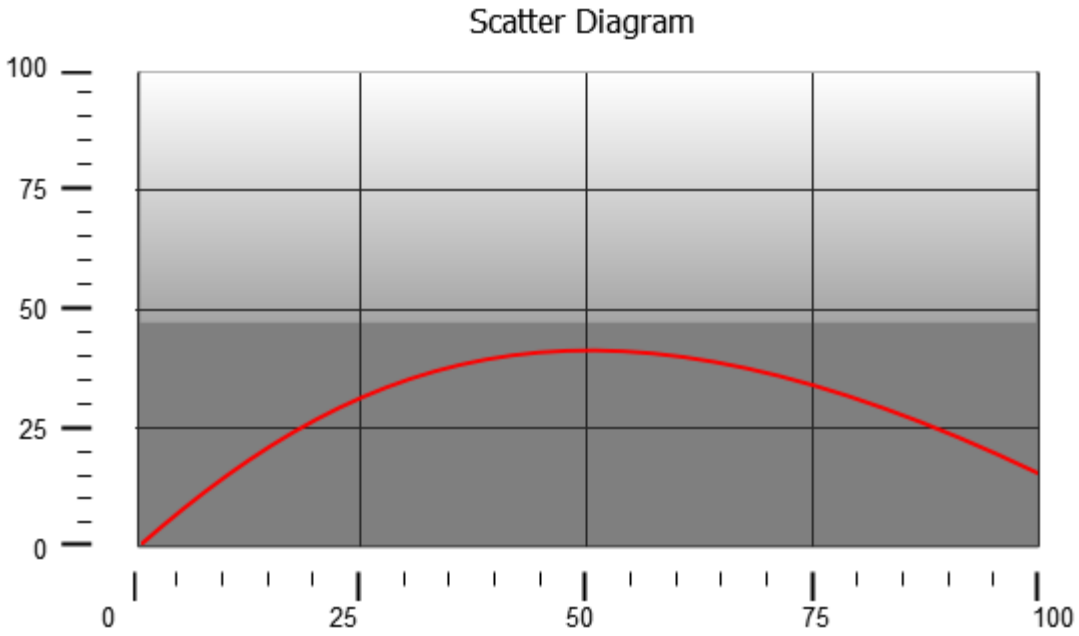


Figure 166

In Scatter Diagram a linear interpolation of points is done.
To create a new Scatter Diagram you have to proceed as follows:

- Add **Scatter Diagram** widget into the page
- Select the number of curves (**Graph1**...**Graph5**) to show
- Customize the general graph properties as for Trends like X Min, X Max, Grid details
- Define the max number of samples/values to consider (**Max Samples)** for each curve. This parameter set the max number of values to show in the graph starting from first element in the array. Ex. Tag1[20] and Max Samples = 10 will show just first 10 elements of the Tag1 array.
- Define for each curve the two Tags of type Array to show (**X-Tag** and **Y-Tag**).

When the array tags change, is possible to force a refresh using the dedicated macro **RefreshTrend.**

NOTE *The ScatterDiagram is considered as a different type of Trend Widget. However only the RefreshTrend macro is supported for it.*

18 Data Transfers

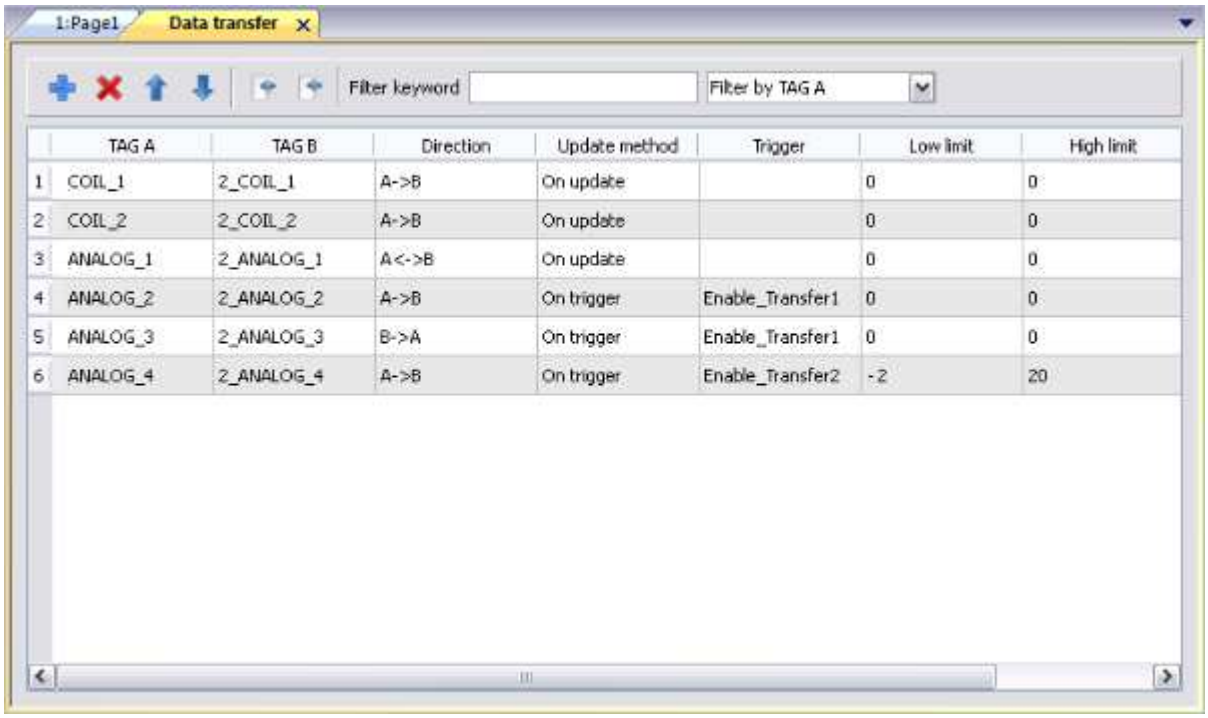
The Data Transfer feature allows the transfer of variable data from one device to another.
Using this feature an HMI panel can operate as a gateway between two devices, even if they do not use the same communication protocol.

18.1 The Data Transfer Editor

To configure each data transfer job, you need to correctly map the tags. This mapping is performed from the Data Transfer editor.

To configure the data transfer:

1. Double click on **Config** node.
2. Double click on **Data Transfer** item.
3. To add a tag, click on the “+" icon: a new tag line is added.



	TAG A	TAG B	Direction	Update method	Trigger	Low limit	High limit
1	COIL_1	2_COIL_1	A->B	On update		0	0
2	COIL_2	2_COIL_2	A->B	On update		0	0
3	ANALOG_1	2_ANALOG_1	A<->B	On update		0	0
4	ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer1	0	0
5	ANALOG_3	2_ANALOG_3	B->A	On trigger	Enable_Transfer1	0	0
6	ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer2	-2	20

Figure 167

Each line in the Data Transfer editor defines a mapping rule for the alignment of the two tags.
You can define more mapping rules if you need different update methods or directions.

18.2 Data Transfer Toolbar Buttons

Import/ Data Transfer settings can be imported and exported in .csv format. This feature can be

Export effectively used whenever it is more convenient to perform changes directly in the .csv file and then reimport the modified file.

Filter keyword Sort only rows containing that keyword. Click on the list box to select the column where you need to apply the filter.

18.3 Data Transfer Fields

TAG A/ TAG B Names of the pair of tags to be mapped in order to be exchanged through the HMI panel.

Direction **A->B** and **B->A**: Unidirectional transfers, values are always received by one tag and sent by the other tag in the specified direction.

A<->B: Bidirectional transfer, values are transferred to and from both tags.

Update Method **OnTrigger**: Data transfer occurs when the value of the tag set as the trigger changes above or below the values set as boundaries of a tolerance range. Limits are recalculated on the previous tag value, the same that triggered the update.

NOTE this method applies only to unidirectional transfers (**A->B** or **B->A**).

OnUpdate: Data transfer occurs whenever the value of the source tag changes.

NOTE this method applies both to unidirectional and to bidirectional transfers (**A->B**, **B->A** and **A<->B**).

NOTE The runtime monitor source tags (the trigger tag when using OnTrigger or tags to transfer when using OnUpdate) for changes in a cyclic way based on Tag Editor “Rate” parameter. So, if rate of source Tag is 500ms (default) system check for updates every 500ms. All changes on source tag < “rate” time are ignored.

on Startup When checked, execute data transfer on startup if quality of source tag is GOOD.

NOTE Data transfers executed on Startup could have major impact on hmi boot time. Avoid to use it where not really needed.

Trigger, High limit, Low limit Tag values that trigger the data transfer process. When this tag changes its value outside the boundaries set as **High limit** and **Low limit**, data transfer is started. The range of tolerance is recalculated according to the specified limits on the tag value which triggered the previous update. No action is taken if the change falls within the set limits.
This mechanism allows triggering data transfers only when there are significant variations of the reference values.

NOTE if both **Low limit** and **High limit** are set to "0", data transfer is triggered as soon as there is a change in the value of the trigger tag.

NOTE **Low limit** is less or equal to zero.
Below an example where:
High limit=1,9
Low limit=- 0,9
• = points where the data transfer is triggered

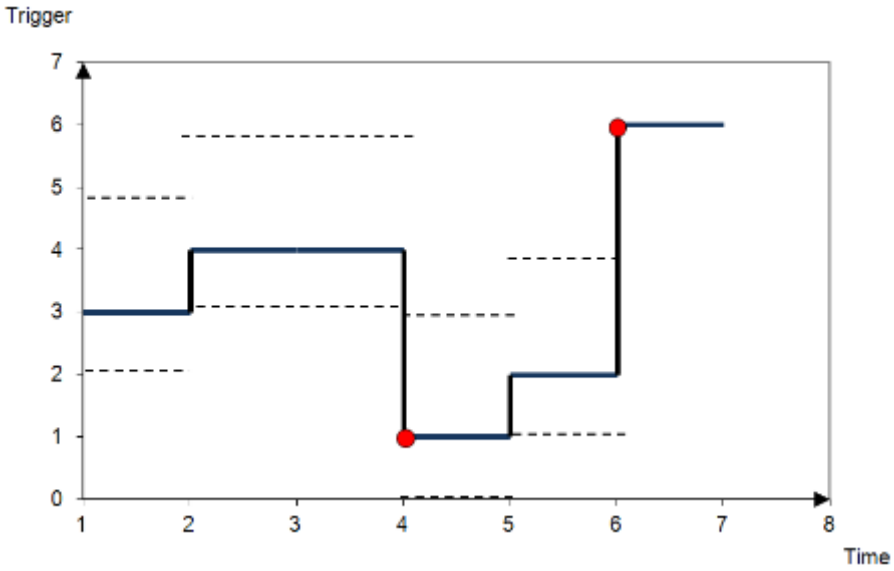


Figure 168

18.4 Exporting Data to .csv Files

Configuration information for data transfers exported to a .csv file. Example is shown in figure below.

A	B	C	D	E	F	G	H	I	J
COIL_1	2_COIL_1	A->B	On update		0	0	data1	true	1
COIL_2	2_COIL_2	A->B	On update		0	0	data2	true	1
ANALOG_1	2_ANALOG_1	A<->B	On update		0	0	data3	true	1
ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer1	0	0	data4	true	1
ANALOG_3	2_ANALOG_3	B->A	On trigger	Enable_Transfer1	0	0	data5	true	1
ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer2	-10	20	data6	true	1

Figure 169

Columns A through G contain the same data as in the Data Transfer editor. Some additional columns are present.

Column H Unique identifier automatically associated by the Data Transfer to each line.
When you edit the .csv file and you add one extra line, make sure you enter a unique identifier in this column.

Columns I-J reserved for future use.

18.5 Data Transfer Limitations and Recommendations

Correct definition of data transfer rules is critical for the good performance of the HMI panels.

To guarantee reliability of operation and performance keep in mind the following rules:

- The **OnTrigger** method allows only unidirectional transfers, (A->B or B->A)

- The **OnUpdate** method allows changing the values in accordance with the direction settings only when the source value changes.
- JMobile Studio is not a supervisory system. Its performance depends on:
    - number of data transfers defined in the Data Transfer editor
    - number of data transfers eventually occurring at the same time
    - frequency of the changes of the plc's variables that are monitored
    - Number and size of features used in the project (i.e. tags, Alarms, Trends…). Always test performance of operation during project development.
    - If inappropriately set, data transfer tasks can lead to conditions where the tags involved create loops. Identify and avoid such conditions.

- Using the "**OnUpdate**" mode of the Data Transfer you force the system to continuously read all the defined source tags to check if there are changes that need to be transferred. Consider to reduce the update rate of the source tags that need to be checked. The default value of the update rate of each tag is 500 mSec and can be modified from the tag editor using the advanced view. To use two different rates for data transfer & Pages, tags can be defined two times in tag editor, one for pages and another for data transfers.
- Data Transfer configured using the OnTrigger mode is preferred over the OnUpdate mode because you have the possibility to force the Data Transfer based on your needs. You can use the scheduler to calibrate the update rate based on the performance of your entire application. Also using OnTrigger system is monitoring just trigger tag and not all tags to transfer.
- Using **tags of type array** allows server engine to optimize data transfer and reduce workload.
- Too many Data Transfers can introduce side effects on page change time and on boot time.

19 Offline Node Management

When one of the devices communicating with the HMI panel goes offline, this may reduce the overall communication performance of the system.

The offline node management feature recognizes offline devices and removes them from communication until they come back online.

Additionally if you know that any of the devices included in the installation is going to be offline for a certain time, you can manually disable it to maximize system performance.

NOTE *This feature is not supported by all communication protocols. Check protocol documentation to know if it is supported or not.*

19.1 Offline Node Management Process

Steps of the process are:

- A certain device is online and it is regularly polled by the system: if the device does not answer to a poll, the system polls it again twice before declaring the device offline.
- When a device is offline, the system polls the device with a longer interval, called Offline Retry Time (ORT). If the device answers to the poll, the system declares it online and starts polling it at regular intervals.

The following schema shows the three polling attempts and the recovery procedure that starts when the Offline Retry Time is elapsed:

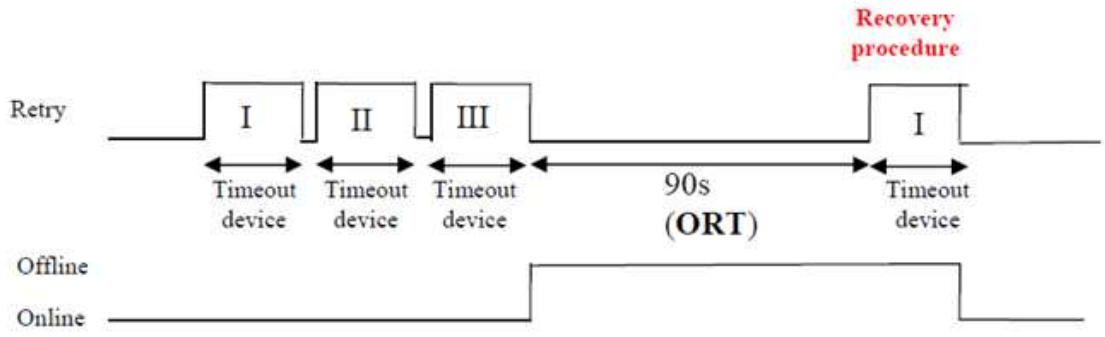


Figure 170

19.2 Manual Offline Node Management Process

Offline Node Management can be done manually.

- a specific device on a Node ID is online and it is regularly polled by the system
- Using a macro action connected to a button the user can declare the device offline: the system stops polling it.
- Another macro action can be used to declare the device online: the system restarts polling it at regular intervals.

19.3 Manual Offline Configuration

When you know that some devices in communication with the HMI are going to remain offline for a certain period of time, you can exclude them from data polling using the **Enable node** macro action.
For example, you can customize your page to contain a button and associate it to an action that will allow you to exclude and/or include a specific device node as needed.

The following example explains how to create a button that, when pressed, will disable an associated device. To do this:

1. In a page of your project add a button.
2. Associate an event to the button (for example **OnMouseRelease**)

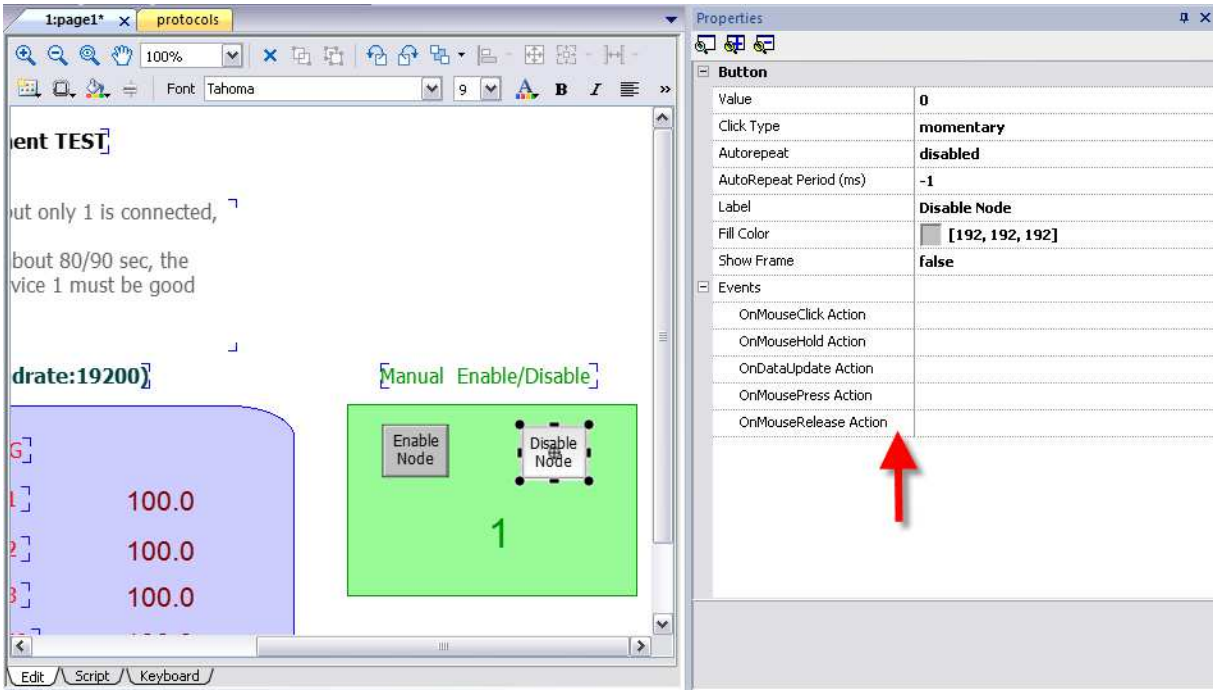


Figure 171

3. Click on the event row, click the '+' button and select **Add action**
4. Add the **Tag Action EnableNode** to the event (**Tag Action** -> **EnableNode**).

5. Make sure that the **Enable** field is set to “false”.

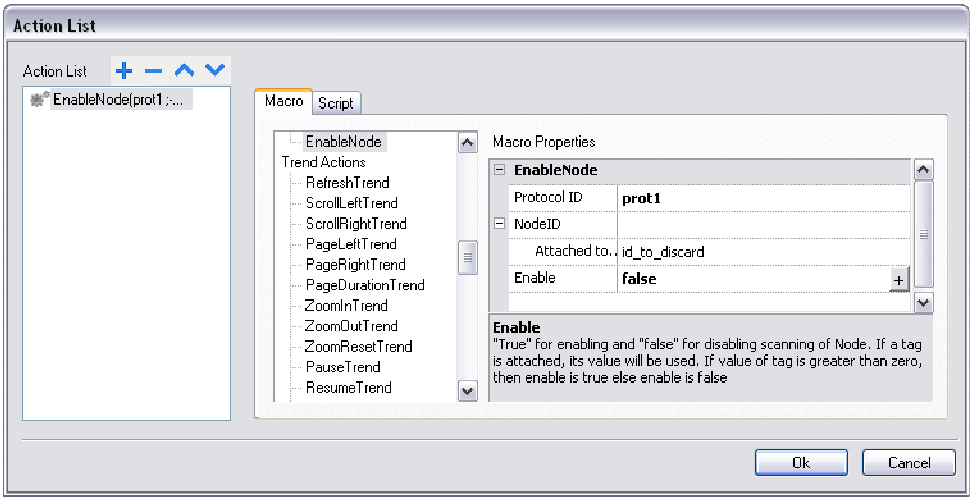


Figure 172

6. Enter the correct **Protocol** and **Device ID** and click **Ok**.

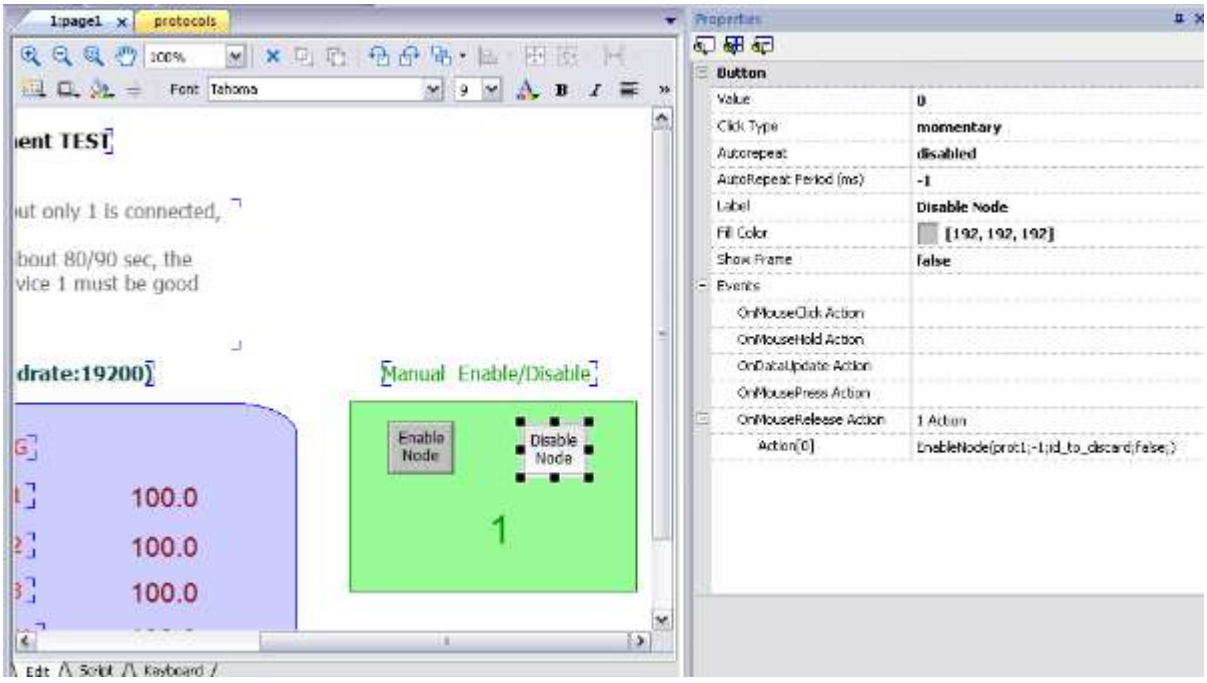


Figure 173

7. The event set is shown in the row. The associated device is indefinitely disabled and therefore no longer polled for data collection.

In the situation described above, you may want to create another button to re-enable the device when needed, in this case the **Enable** field will have to be set to “true”.

WARNING: all disabled device nodes will remain disabled if the same project is downloaded on the panel, on the other hand, if a different project is downloaded, all disabled devices will be re-enabled. The same happens on package update.

Tip: *to make this feature more dynamic, you may decide not to indicate a specific NodeID but attach it to the value of a tag or to an internal variable created to identify different devices that might be installed in your network.*

NOTE *when using the action* ***Enable Node*** *described above to force a device node back online, data polling will start immediately.*

19.4 Automatic Offline NodeDetection

HMI panels can automatically disregard connected devices which are found to be offline.
When a device is found offline the first time, it is polled twice before being disregarded.
When it is declared offline it is polled at different intervals that can be set by the user.

To set the offline polling on one node ID:

1. Click the **Config** node and click **Protocols**.
2. Select the desired node ID.
3. Click on the **Show Advanced Properties** button: more columns are added to the table.
4. In the table set the **Offline Retry Time** parameter: the device on this node ID will be polled with this frequency when offline.

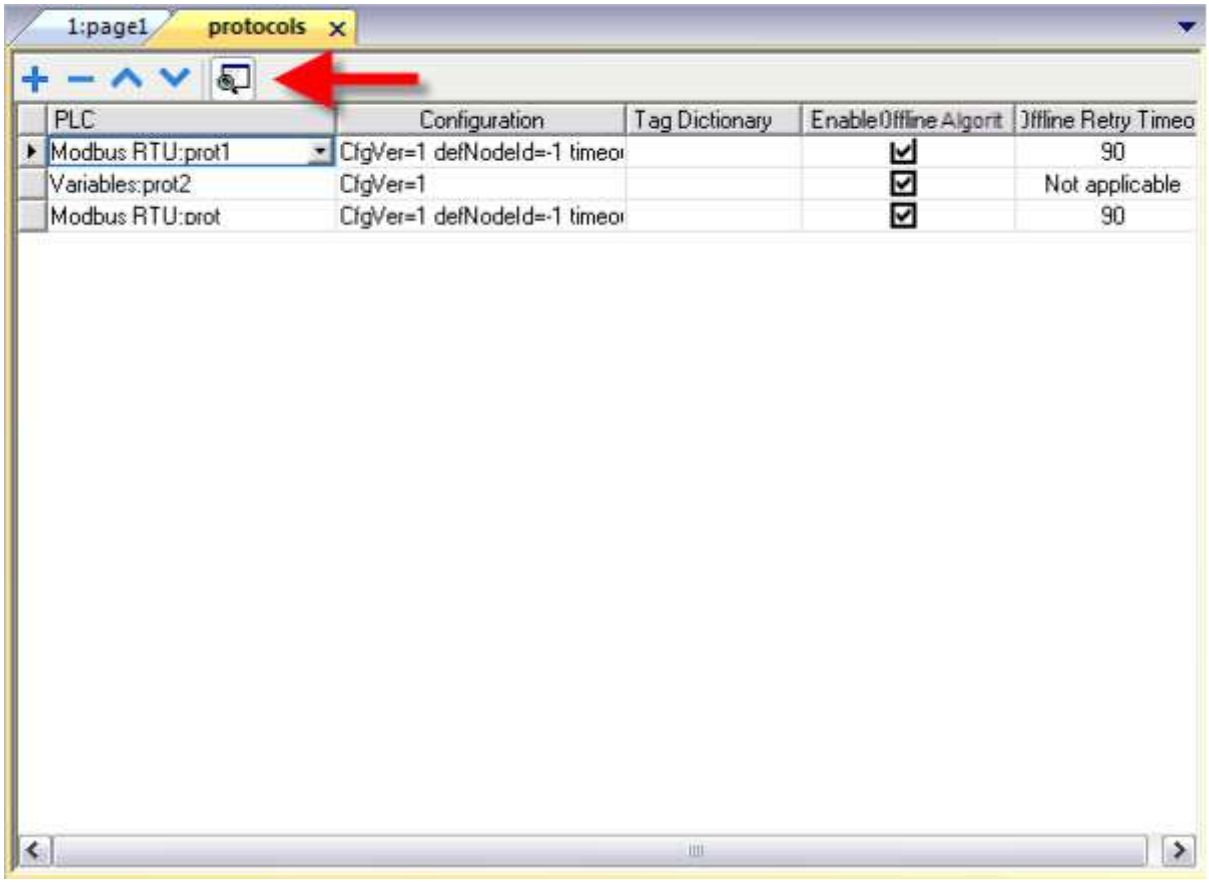


Figure 174

19.5 Offline Management Toolbar buttons

Advanced properties It shows/hides the advanced properties columns.

19.6 Offline Management Fields

PLC	Protocol type and name.
Configuration	Protocol settings.
Tag dictionary	Tags imported for the protocol
Enable Offline Algorithm	Enable the Offline Management for the protocol
Offline Retry Time	Interval, expressed in seconds, between when the node was disregarded and when the recovery procedure started. Max value for ORT is 86400sec (24h).

20 Multi-Language

A true Multi-language feature has been implemented in JMobile Studio through code pages support from the Microsoft Windows systems. The Multi-language feature handles different code pages for the different languages. A code page (or a script file) is a collection of letter shapes used inside each language.

The Multi-language feature can be used for a project by defining languages and character sets. JMobile Studio also extends the TrueType Fonts (in short TTF) provided by Windows systems to provide different font faces associated with different character sets.

JMobile Studio has features that allow users to provide strings for each of the languages.
When in edit mode, JMobile Studio provides support to change the display language from a language combo box. This helps users see the page look and feel at design time.

NOTE *In Windows XP operating systems, for the proper operation of the Multi-language editor in the Studio, you will need to install the support for complex script and East Asian languages as shown in the figure below.*

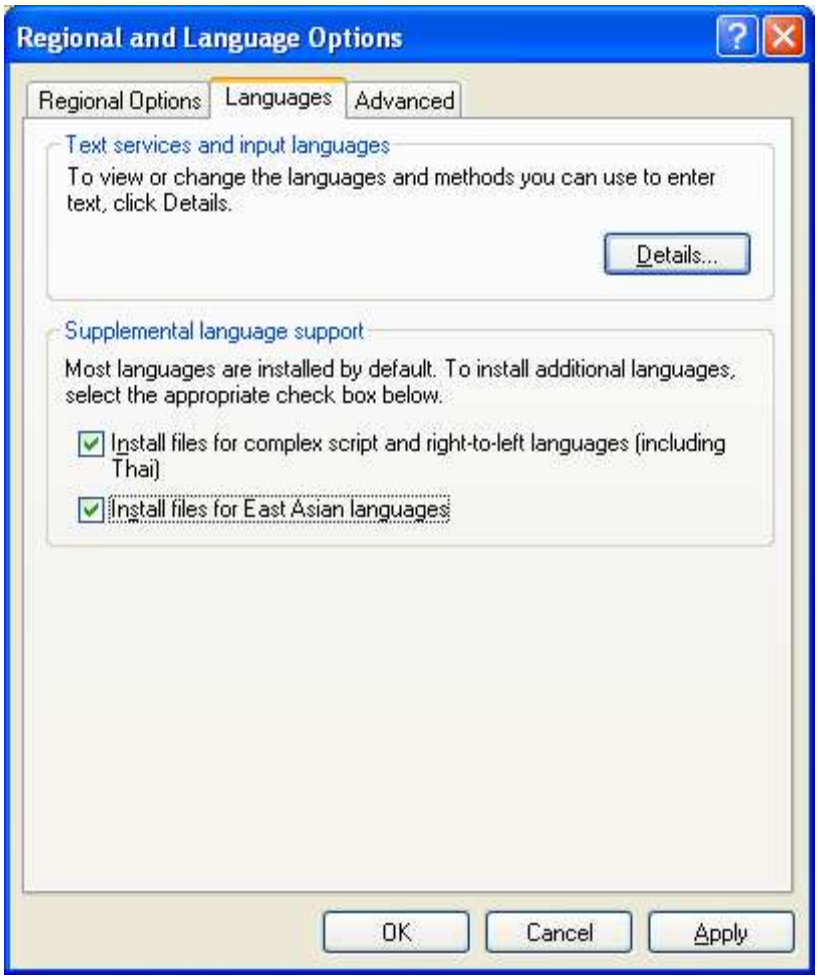

Regional and Language Options
Regional Options
Languages
Advanced
Text services and input languages
To view or change the languages and methods you can use to enter text, click Details.
Details...
Supplemental language support
Most languages are installed by default. To install additional languages, select the appropriate check box below.
Install files for complex script and right-to-left languages (including Thai)
Install files for East Asian languages
OK
Cancel
Apply

Figure 175

JMobile Studio is actually supporting a restricted set of fonts for the Chinese languages.

For Simplified Chinese, following fonts are supported:

Fangsong	simfang.ttf
Arial Unicode MS	ARIALUNI.TTF
Kaiti	simkai.ttf
Microsoft Yahei	msyh.ttf
NSImsun	simsun.ttc
SimHei	simhei.ttf
Simsun	simsun.ttc

For the Traditional Chinese, following fonts are supported:

DFKai-SB	kaiu.ttf
Microsoft Sheng Hai	msjh.ttf
Arial Unicode MS	ARIALUNI.TTF
MingLiU	mingliu.ttc
PMingLiU	mingliu.ttc
MingLiU_HKSCS	mingliu.ttc

20.1 Add a Language to Project

To add a language to a project, launch Multi-language from the Project View pane. Click the “Add" button to add the language, then select the Writing system and the Default Font used by all the “table like" widgets (such as alarms or events). Use the “Default" button to set the default language used when the Runtime starts Multi-language.

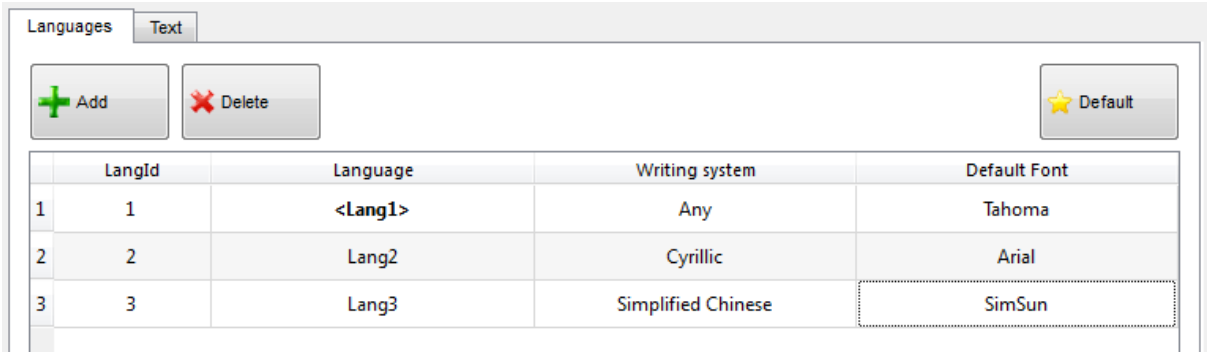


Figure 176

20.1.1 Language Display Combo

This combo can be used to change language at the design phase. This helps users to view the page in the different supported languages at design time itself.

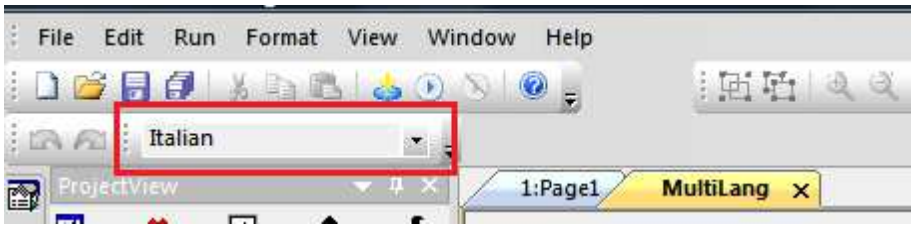


Figure 177

20.2 Multi-Language Widget

Multi-language support is available for different objects, like push buttons, static text, message, alarm description and pop-up messages.

20.2.1 Multi-Language for Static Text Widget

When you double click a text widget on the page, the dialog shown below will open. Here, you can edit the text for the selected language and select the font.
The bold, italic and color properties are set for all the languages globally for the widget. Text for each of the languages can be given, by selecting the language from the combo box. However, it is recommended that you use the export and import features, as described in the chapter "Export and Import of Multilanguage Strings".

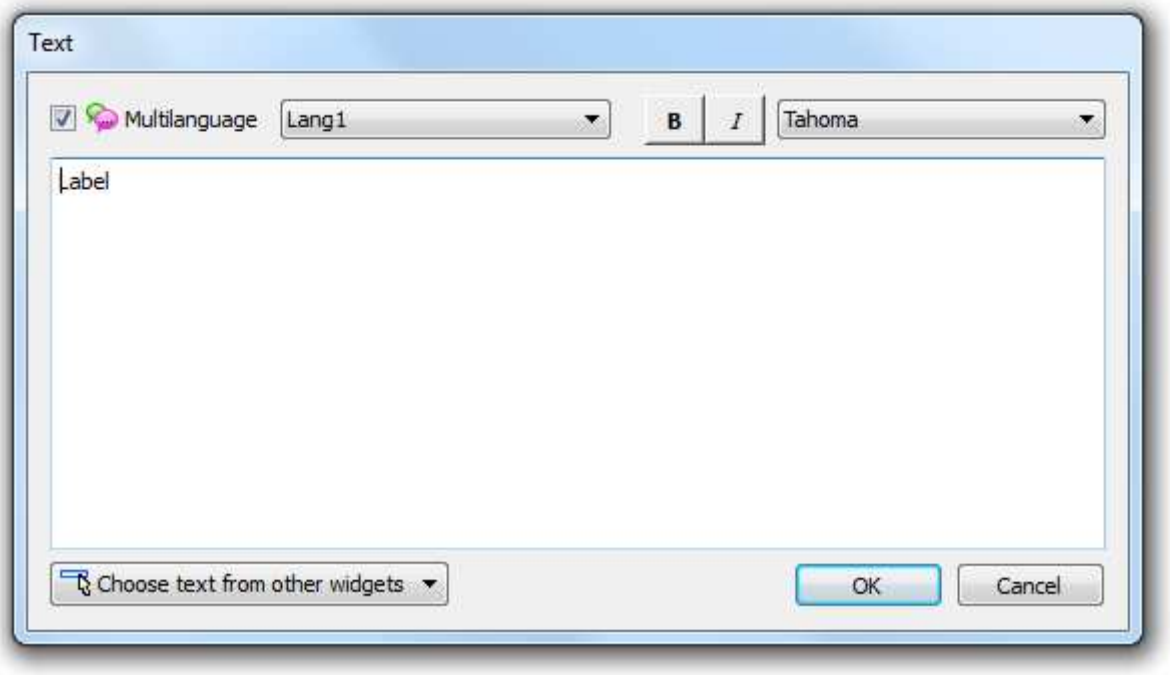


Figure 178

20.2.2 Multi-Language for Message Widget

JMobile Studio allows you to use Multi-language in the message widget. After you drag and drop a message widget, select the language from the Language combo box and enter the message description for the selected language. . Again, you can also use the export and import features, as described in the chapter" Export and Import of Multilanguage Strings".

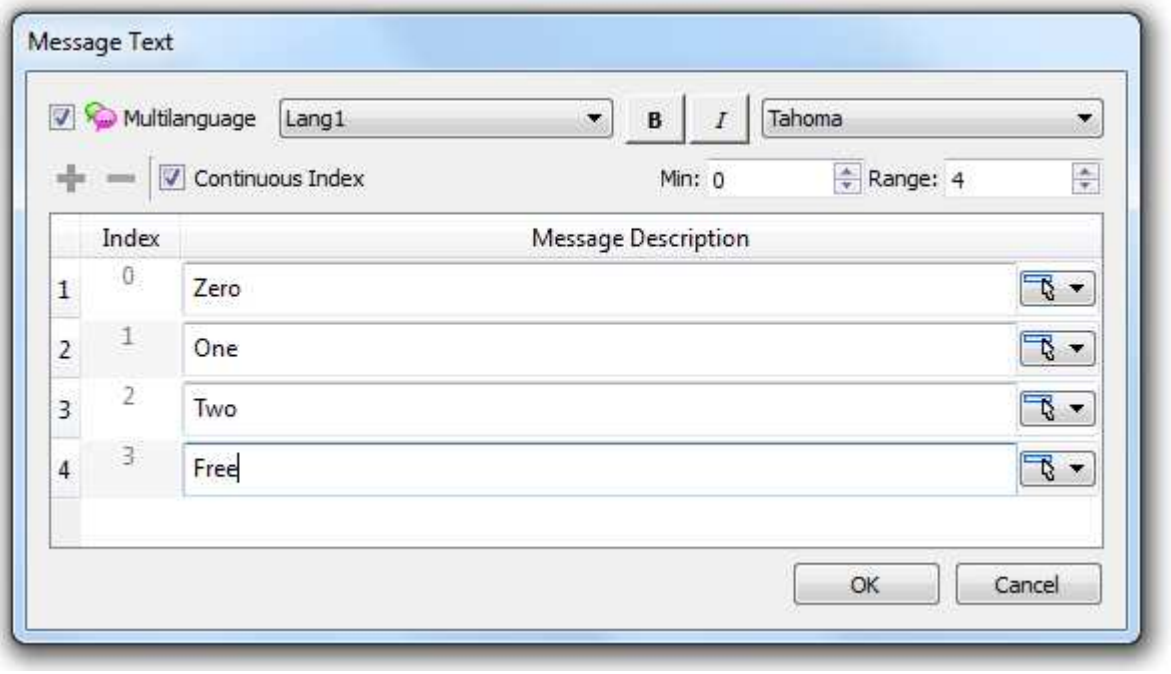


Figure 179

20.2.3 Multi-Language for Alarm Messages

JMobile Studio allows you to use Multi-language for Alarm messages.
To add a Multi-language string for an Alarm message, open the alarm editor, select the language list from the tool bar (Language combo) and add the alarm messages. You can also use the export and import features, as described in the chapter "Export and Import of Multilanguage Strings".

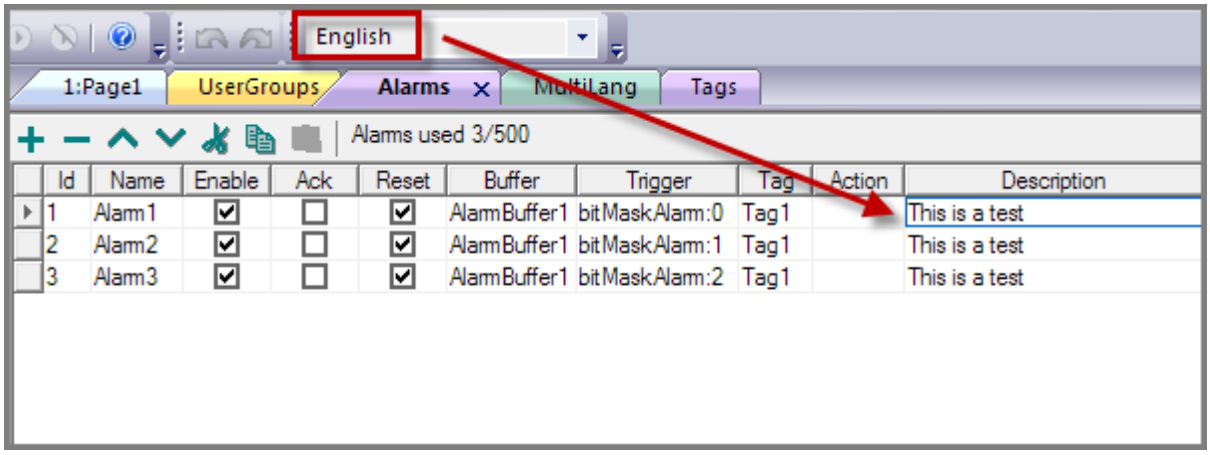


Figure 180

20.2.4 Multi-Language for Pop-up Messages

For the popup message macro, you can define the Multi-languages. To do this, you first need to select the language from language list combo, and then enter the message in the Show Message macro (as shown in the figure below).

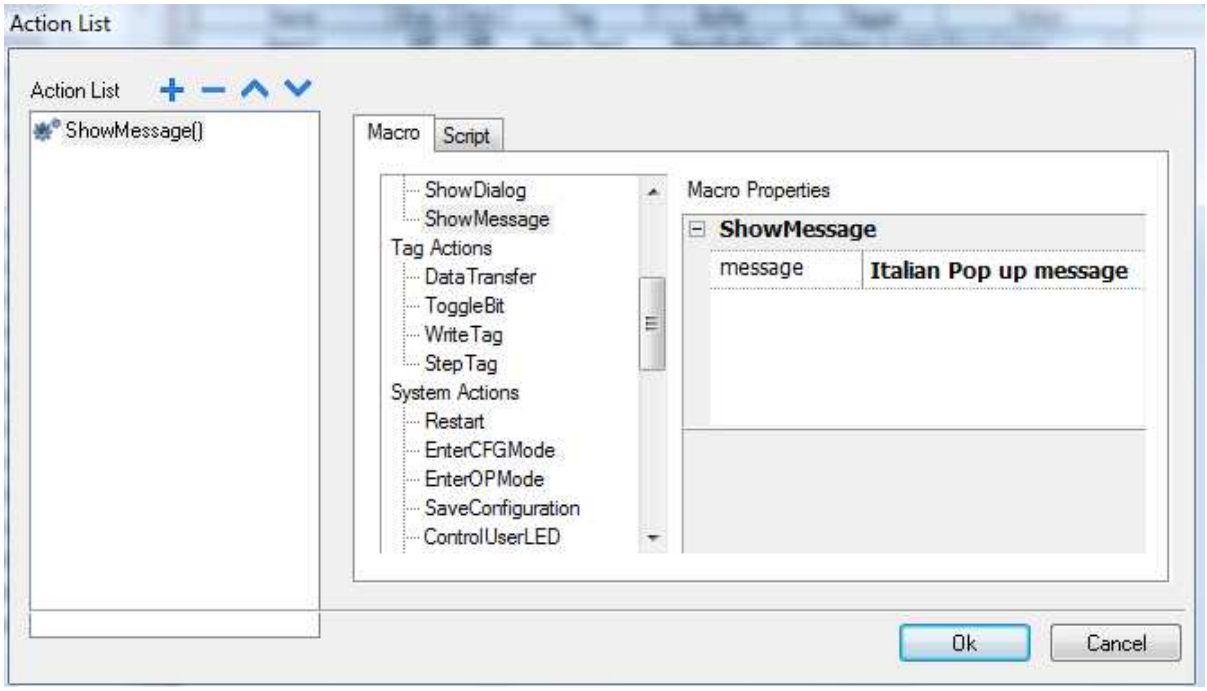


Figure 181

20.3 Export and Import of Multi-language Strings

The easiest way to translate a project into multiple languages is to use the **Export** feature, exporting all text to a file. The translation can be done in that document, then using the **Import** feature, brings all text for all languages back into the project.
The Multi-language strings will be exported in CSV file format, then you can modify the strings with an external editor, and import it back to the JMobile Studio.
The CSV file exported by JMobile Studio is coded in Unicode. To edit it, you need a specific tool that supports CSV files encoded in Unicode format.
To export the Multi-language strings, open the Multi-language editor and switch to **Text** view. Then, click the **Export** button and save the CSV file. You can then modify the exported CSV file and **Import** back to JMobile Studio. Click the **Save** button to save the text.

NOTE *It is recommended that you set all languages that will be used in the project before exporting the file. This will guarantee that the exported file will contain all columns and language definitions for that project.*

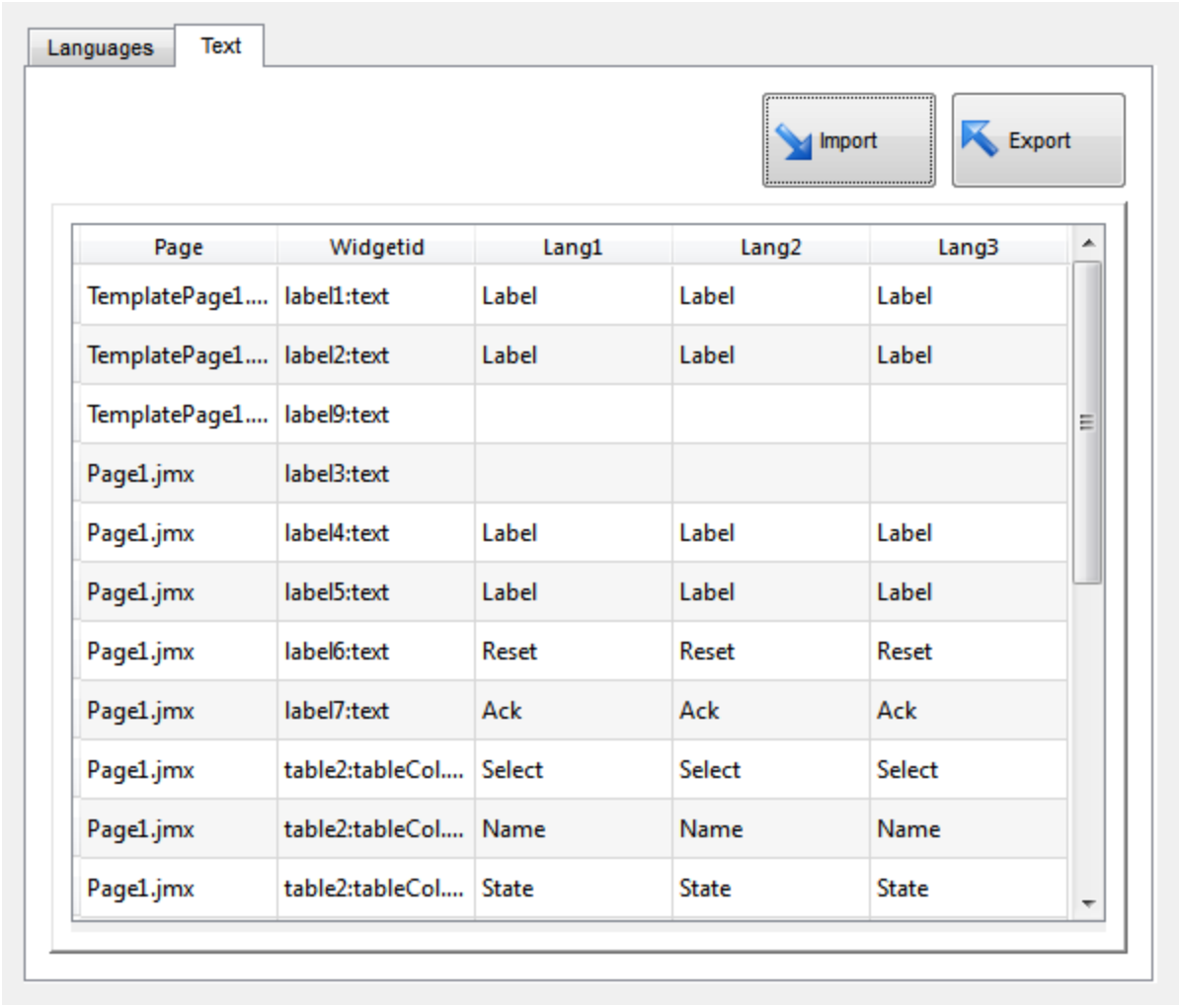


Figure 182

The strings are imported matching the widget ID and the page number of each widget.
To change the separator used in the exported file, please have the regional settings of your work PC changed. Upon importing, the separator information is retrieved from the file; if not found, the default character "," is used. Immediately after the Import, the modified strings will be displayed in the text tab. Once the user hits the button to "Save" the changes, the changes are saved to the internal widgets.

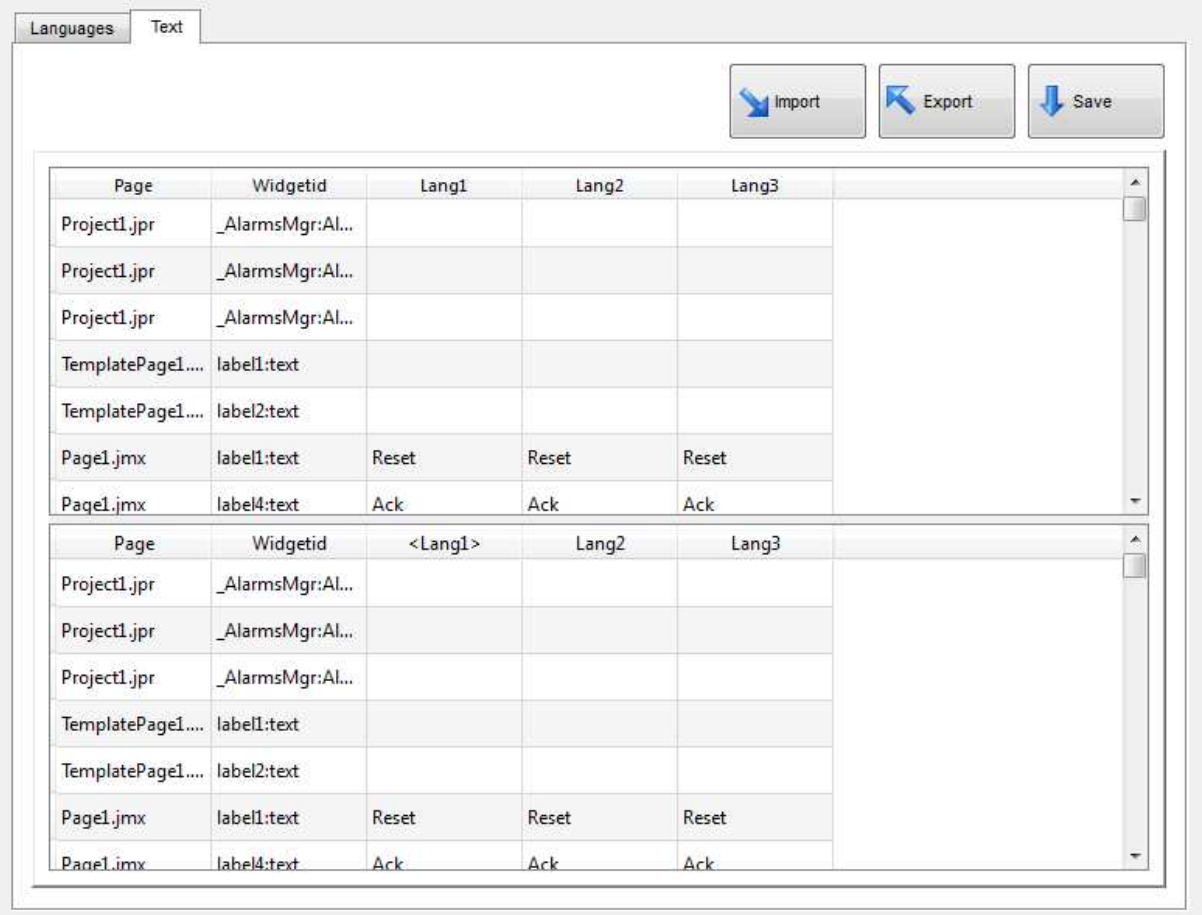


Figure 183

The feature **Import** supports two formats (Figure 184):

- **Comma Separated Values** (.csv)
- **Unicode Text** (.txt)

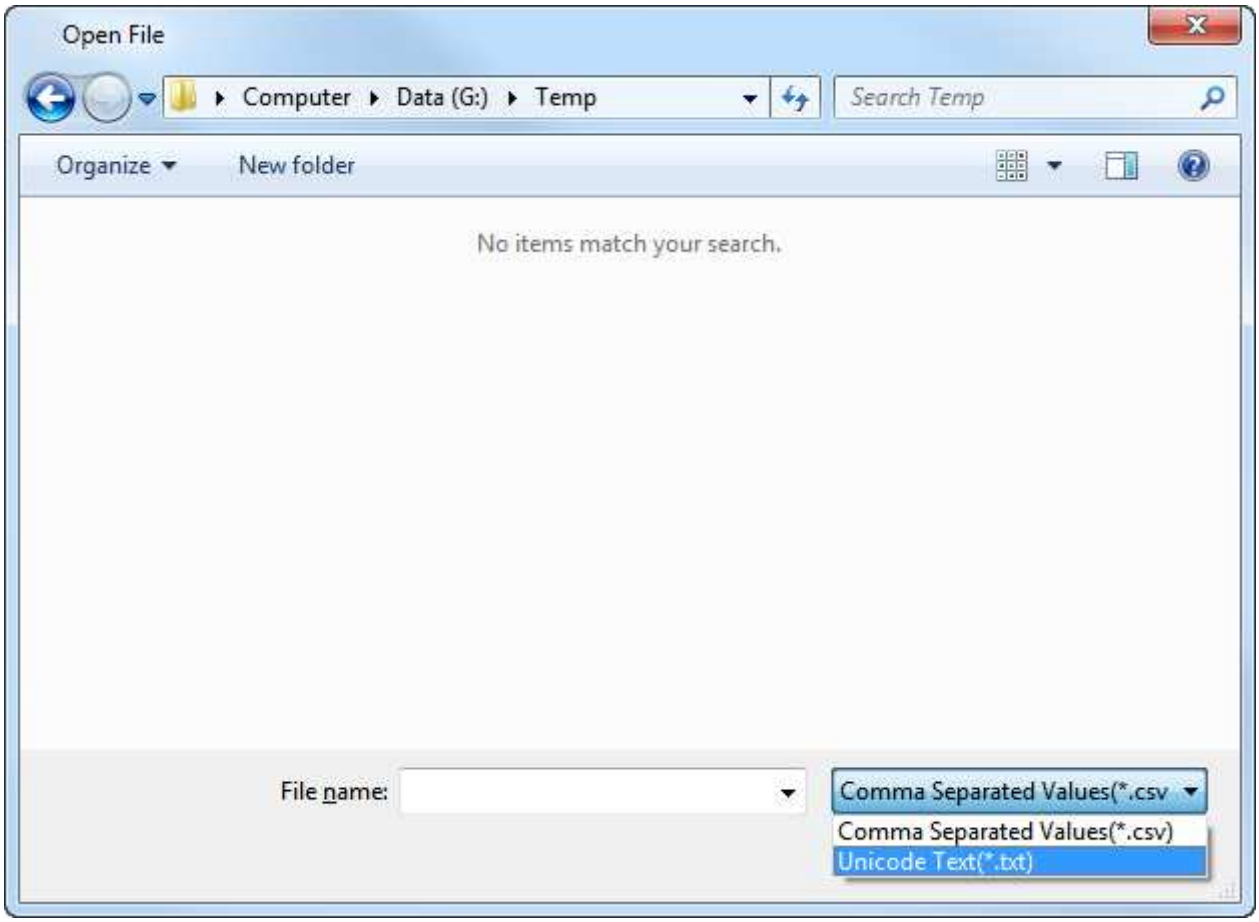


Figure 184

The **Unicode Text** file format must be used every time you import a file modified by Microsoft® Excel®. You can save your Excel® sheet in this format choosing **File > Save As…** and choose the option **Unicode Text (*.txt)** from the **Save as type:** combo.

20.4 Change Languages at Runtime

After the project download, Runtime will start using the Default Language. However, you can change the language on Runtime using the “SetLanguage" macro.

LangID is the language index corresponding to the language ID, as it can be read from the Language Configuration Editor.

NOTE *After languages are changed at runtime with the macro execution, the current language is saved and retained for the next run.*

20.5 Limitations in UNICODE support

JMobile Studio has been designed for working with UNICODE text. However, for compatibility reasons with all platforms, UNICODE is supported only in a subset of field types.

Area	Field	Charset Accepted	Reserved Chars/Strings
Protocol Editor	Alias	ASCII [32..126]	(space) , ; : . < * >'
Tag Editor	Name	ASCII [32..126]	. \ / * ? : > < | " & # %;=
	Group	ASCII [32..126]	<New> \ / * ? : > < | " & # % ;
	Comment	Unicode	
Trends	Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
Printing Reports	Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
Alarms	Name	ASCII [36..126]	\ / * ? : > < | " & # % ;
	Description	Unicode	[] - for live tags, \ escape seq for [and \
Events	Buffer Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
Scheduler	Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
Languages	Language Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
	Texts in widgets	Unicode	
	Texts from import files	Unicode	
User Group	Group Name	a-z A-Z _	admin,guest,unauthorized
	Comments	Unicode	
User	Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
	Password	Unicode	
	Comment	Unicode	
Recipes	Name	ASCII [32..126]	\ / * ? : > < | " & # % ; ! $'()+,=@[]{}~`
	Set Name	ASCII [32..126]	\ / * ? : > < | " & # % ; ! $'()+,=@[]{}~`
	Element name	ASCII [32..126]	\ / * ? : > < | " & # % ; ! $'()+,=@[]{}~`
General	Project Name	A-Z,a-z,0-9,-,_	“PUBLIC”, “readme”, “index.html”
	Page Name	A-Z,a-z,0-9,-,_	
	Dialog Page Name	A-Z,a-z,0-9,-,_	
	Template Page Name	A-Z,a-z,0-9,-,_	
	Keypad Name	A-Z,a-z,0-9,-,_	
	Files (Images/Video/etc..)	A-Z,a-z,0-9,-,_	
	Widgets ID	A-Z,a-z,0-9,-,_	
Runtime	PLC Communication	UTF-8, Latin1, UCS-2BE, UCS-2LE, UTF-16BE, UTF-16LE	

21 Scheduler

JMobile Studio provides a scheduler engine that can be easily configured to program the execution of specific actions at repeated intervals, or on a time basis.

Depending on your application, creating a schedule is typically performed with a 2-step process:

1. The first step is to define the parameters of the schedule to run on the panel. This includes selecting the actions to perform when the scheduled event is activated. The first step is performed using the Scheduler Editor.

2. The second step is to create a Runtime user interface that allows the end-user to change settings per each defined scheduler. For example, the Runtime user interface will allow the user to turn on a device at 5:00 pm, and turn the device off at 10:00 pm, every day. This can be done by dragging and dropping a predefined Scheduler widget, from the Gallery, and placing it on the page. Once on the page, you can set the properties of the individual GUI elements to create the desired interface to be presented to the end-user.

21.1 Configuring the Scheduler Engine

The configuration of the Scheduler Engine is done using the Scheduler Editor. The Scheduler Editor is accessible from the ProjectView pane (as shown in the figure below).

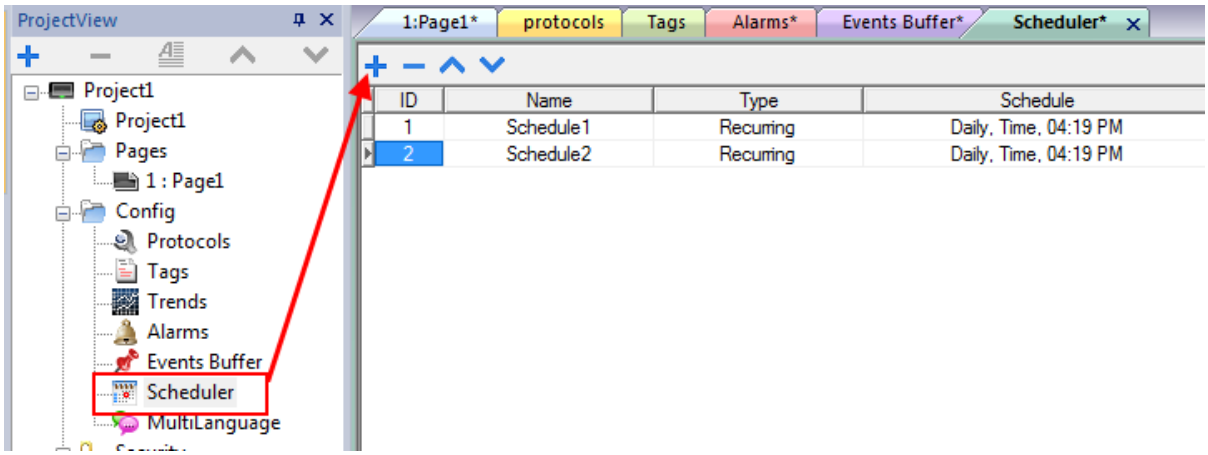


Figure 185

Click on the “+" symbol to add a schedule item. Schedule items can be of two different types as listed below and shown in the figure below:

- **Recurring** Scheduler
- **HighResolution** Scheduler

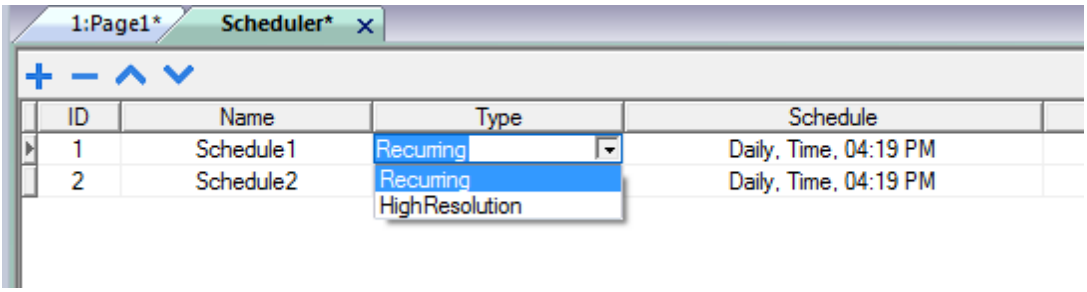


Figure 186

Name
Allows you to define the name of the Scheduler.

Type
Allows you to select the type of Scheduler.

Schedule
Allows you to select different Scheduler options, which are described in chapters Recurrence Scheduler and Type.

Action
Allows you to define macros, which have to be executed at the scheduled time.

Priority
Allows you to set a priority level for the event. This is used in case two distinct schedules occur at the same time. The event with the higher priority will be executed before those of lower priority.

21.2 HighResolution

The HighResolution scheduler can be programmed to perform an action, or sequence of actions, repeatedly, at a specific duration. The High Resolution scheduler can be set in milliseconds. To configure the HighResolution scheduler, select “HighResolution" from the Type column and set the desired duration from the schedule column.

Figure 187

NOTE *The HighResolution scheduler cannot be changed during Runtime. If the user needs to change the schedule Runtime, then the Recurrence scheduler should be used by selecting “Every", which is described in the following chapters. The minimum time resolution, when using a Recurrence scheduler in “Every" mode, is one second.*

21.3 Recurrence Scheduler

The Recurrence Schedulers can be programmed to perform an action, or sequence of actions, and the schedule can be modified during Runtime.

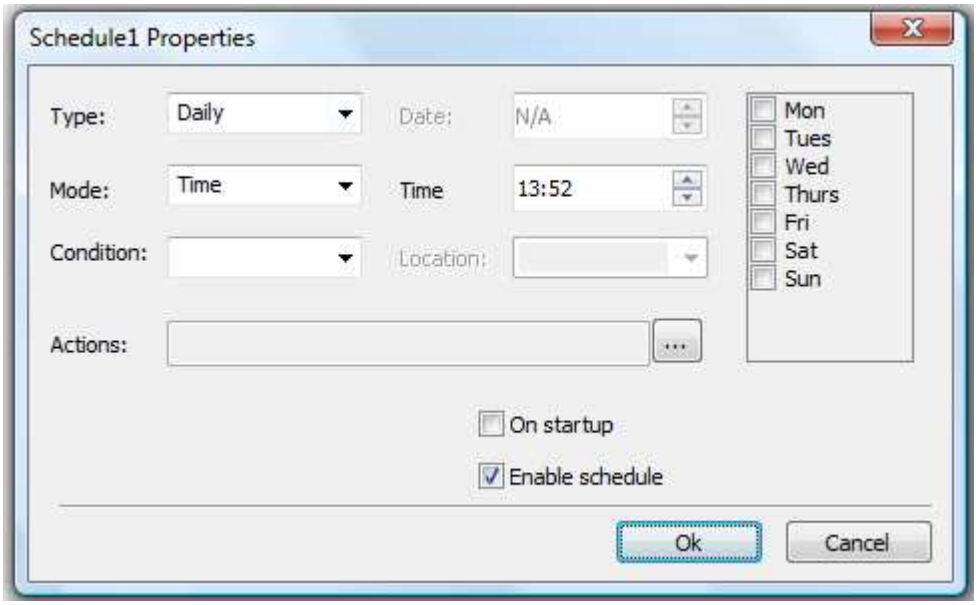


Figure 188

By default, when a schedule is added, the “Enable schedule” checkbox is marked. You have the option to keep a schedule in the project but disable it by unchecking the box.

Each Scheduler can be configured to run once at Startup (when the “On Startup” check-box in marked). Additionally, you can specify the scheduler to be enabled only at the first start up run by using the “Execute only at startup” check box.

Type
The Type combo allows you to select the type of Schedulers (as shown in the figure below). However, you can change the type of scheduler at any time during the Runtime, as described in the chapter: Schedule the Events during Run-time.

Options available for Type are the following:

By Date	By Date scheduler allows you to define the schedule for the specific date and time when the actions shall be executed.
Daily	Daily schedules define the execution of a set of actions on a daily basis by specifying the time of day in which the actions are to be executed.
Every	The Every Scheduler is much like the High Resolution scheduler, with the ability to change it in Runtime. The “Every" Scheduler allows you to execute macros with a specific time interval. The time interval can be set from 1 sec to 1 day.
Hourly	The Hourly Schedules allow you to execute a set of actions on an Hourly basis, by specifying the minute in which the actions have to be executed.
Monthly	The Monthly Schedules allow you to execute a set of actions on a Monthly basis, by specifying the day in which the actions have to be executed.

Weekly	Weekly schedules allow you to execute a set of actions on a Weekly basis by specifying the time and day(s) in which the actions have to be executed.
Yearly	The Yearly schedule allows you to execute a set of actions once a year, specifying the date and time in which the actions have to be executed.

Mode
Mode parameter is available for a subset of scheduler types. It is not supported by scheduler of type Every, Hourly. This parameter allows choosing between following way of working:

Time	This is the default. In this case is needed to specify details about time/date/week. Parameters depend on **Type** of scheduler selected.
Random10	Executed 10 minutes before/later the time specified. So, if time is 10:30, actions is executed in range 10:20...10:40 where 20...40 is random.
Random20	Executed 20 minutes before/later the time specified. So, if time is 10:30, actions is executed in range 10:10...10:50 where 10...50 is random.
Sunrise+	Executed n minutes/hours after sunrise time based on a specific location as explain in next chapter.
Sunrise-	Execute n minutes/hours before sunrise time based on a specific location as explain in next chapter.
Sunset+	Executed n minutes/hours after sunset time based on a specific location as explain in next chapter.
Sunset-	Execute n minutes/hours before sunset time based on a specific location as explain in next chapter.

21.4 Configuring Location in JMobile Studio

In JMobile Studio there is a unique scheduler feature based on sunrise and sunset. Before you start the sunrise or sunset scheduler, you need to define the location. Based on the UTC location, the system automatically calculates the sunrise and sunset time.

In the installation, only a few locations are set by default. If your location does not show up in the list, you can add your location by entering the latitude, longitude and UTC information in the “Target_Location.xml" file located in the JMobile Suite\studio\config folder.

For example, the information for the city of Verona is shown below:

```
<file city="Verona" latitude="45.44" longitude="10.99" utc="1"/>
```

After entering the location information, the software displays the city name in the Location combo list, and you can see the sunrise and sunset time on the dialog (as in the figure below).

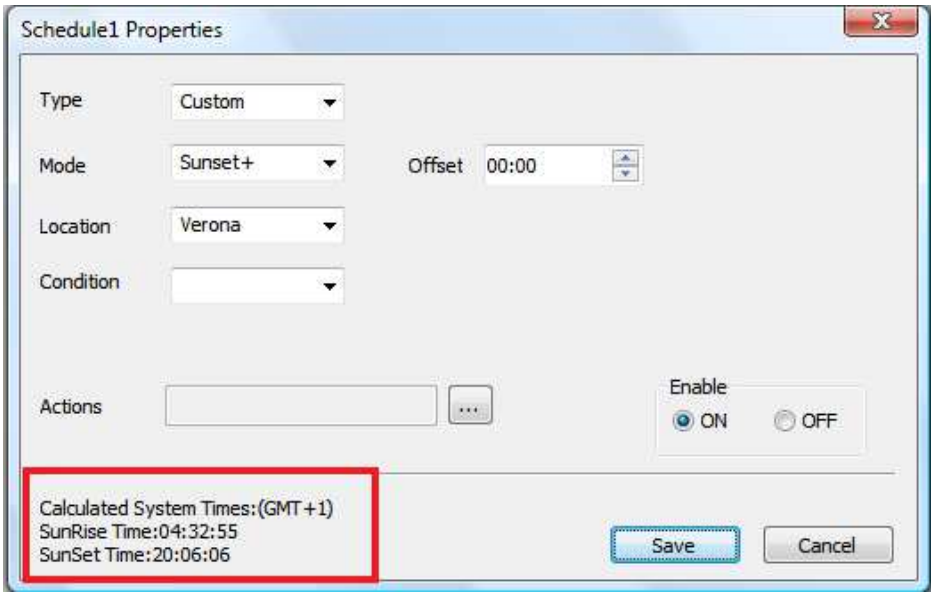


Figure 189

Condition
The Condition combo allows you to select a Boolean Tag (Yes/No) to be evaluated, before activating the specified actions, at the moment the timer is triggered. If Tag = True, actions will be executed, and if Tag = False, the actions will not be executed.
By default, there is “none" => the actions are executed when the timer is triggered.

NOTE *The condition combo will list only the Tag attached to the Boolean data type.*

Actions
From the Action List dialog, you can add as many Actions as desired. The Actions will automatically be executed when the Schedule time occurs.

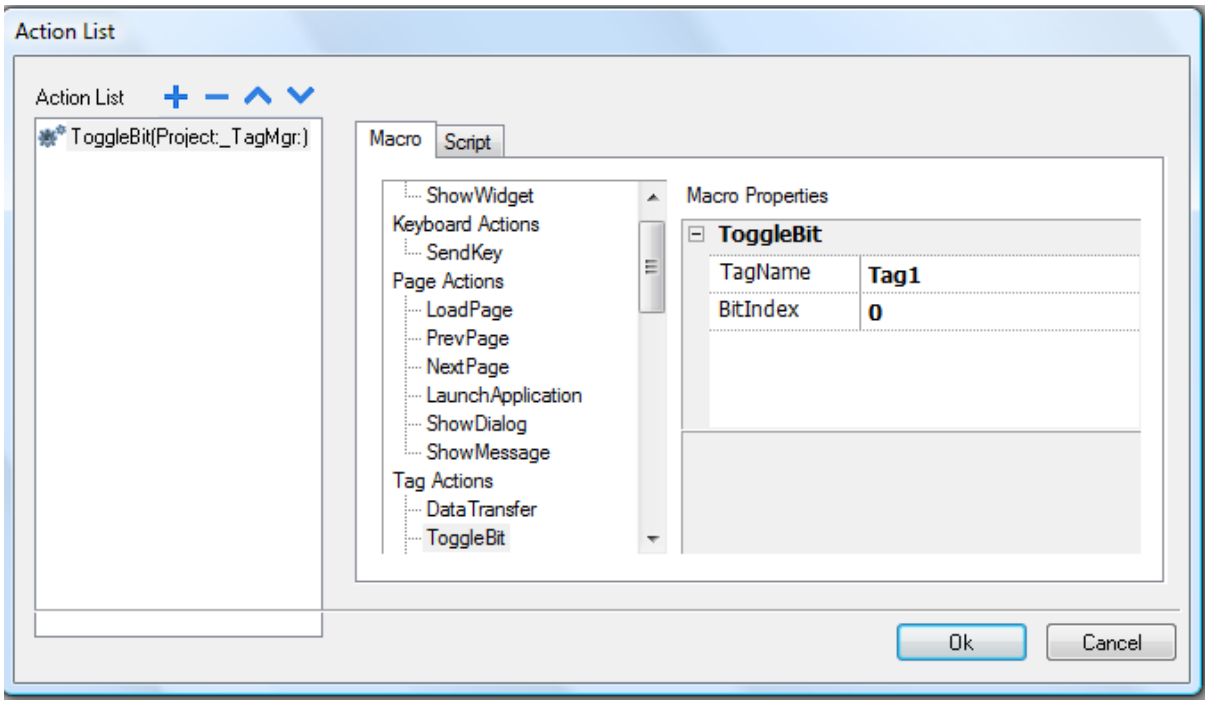


Figure 190

NOTE *The Actions should be programmed in the Studio. Actions cannot be modified at Runtime, all other scheduler parameters can be modified in Runtime (such as, type, mode, location, etc.)*

21.5 Configuring the Schedule Interface for Runtime Interaction

The User Interface for Runtime is the Widget called Scheduler. To add this to the project, just drag and drop it from the advanced section of the Widget Gallery. Once the object is on the page, in order to select the Scheduler items to be displayed in the Widget, click on the + button of the “Name" property that is part of the Scheduler object. A Dialog page will open (as shown in the figure below) where you can add the schedule from the list at Runtime.

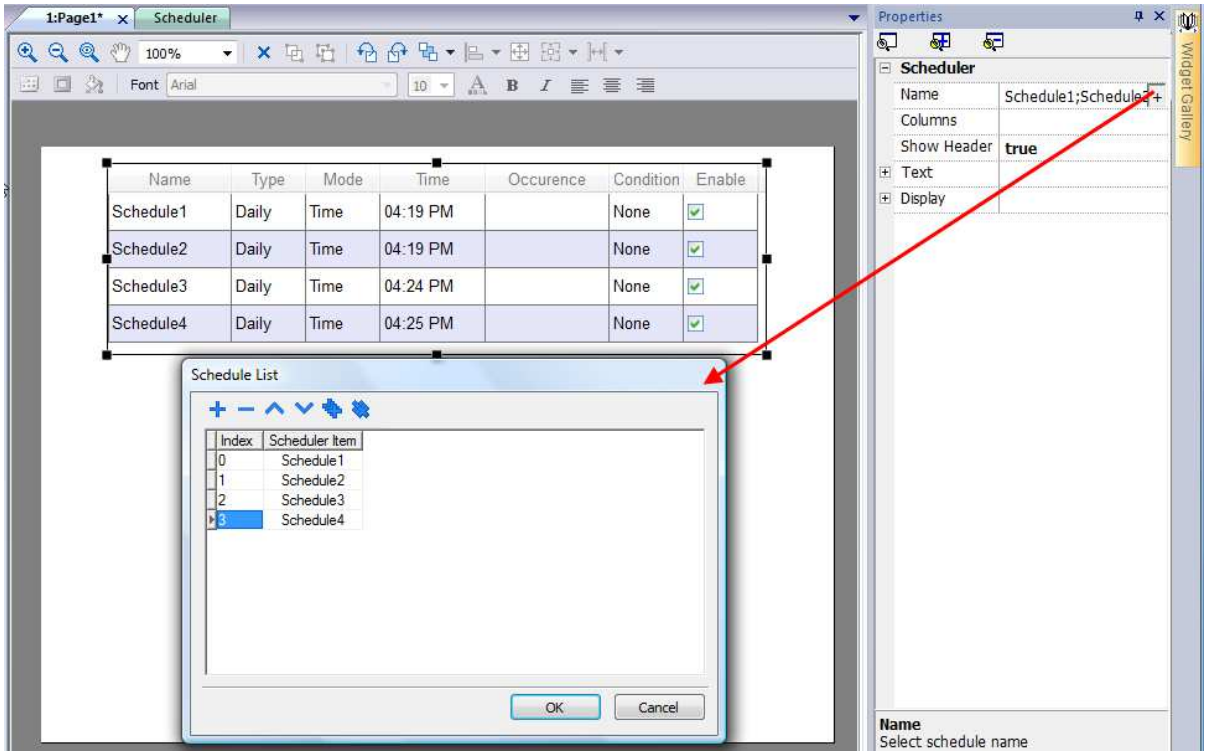


Figure 191

In the Properties pane, you can customize the scheduler Widget to adjust row colors, column width, and show or hide column, etc.

21.6 Schedule the Events during Runtime

If you defined the scheduler GUI on a page (as described in the chapter “Configuring the Schedule Interface for Run-time Interaction”), then you can schedule the event, and modify this schedule, during Runtime on the server.

In Runtime, the user has the flexibility to change all possible types and change the possibility to modes as described in the dedicated chapter.

Name	Type	Mode	Time	Occurence	Condition	Enable
Schedule1	By Date	Time	11:01	JUN 20,2013	None	☑
Schedule3	Monthly	Sunrise+	11:01	Day : 3	None	☑
Schedule4	Weekly	Rando...	16:19	M T W T F S S	None	☑
Schedule5	Yearly	Time	01:00			
Schedule6	Custom	Time	01:16			

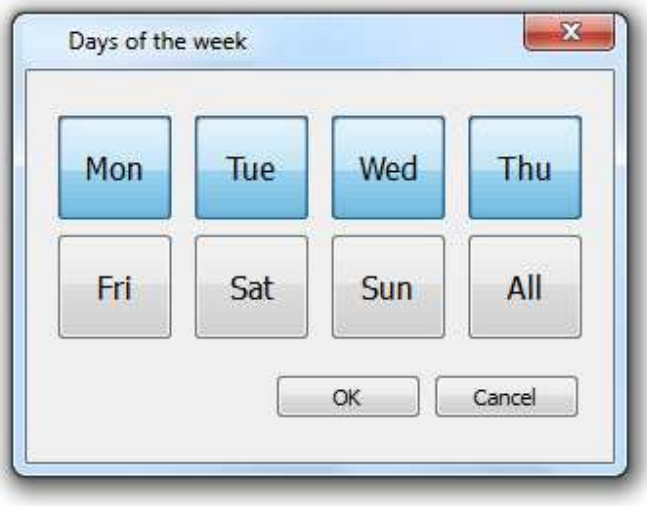


Figure 192

Occurrence
The Occurrence column specifies the date selected by the type of column, as shown in the figure.

Condition
The Condition column lists the available Boolean Tags from the project. If a Tag is selected as a condition, then the scheduler will trigger only when the condition Tag value is 1, otherwise the scheduler will not trigger.

Enable
The Enable check box allows you to enable or disable the schedule. The scheduler will trigger when the enable check box is set. If you want to disable the scheduler temporarily, then uncheck the Enable check box.

22 User Management and Passwords

This chapter describes the user management system. The main purpose of User management module is to restrict access to various objects/widgets and/or operations, by configuring user groups and their authorization level. Users, user groups and authorizations are the 3 entities defined for user management handling.

The basic entity is the user, representing an individual that has the need to work with the system.
Each user must be a member of a group. Users can be a member of just one group. Each group will have different types of authorizations and permissions assigned to them.

Authorizations and permissions for the groups are divided in two basic categories:

- **Widget permissions**: hide, read only, full access
- **Action' permissions**: allowed or not allowed.

The proper combination of these groups and permissions will implement the required level of security options for the application.

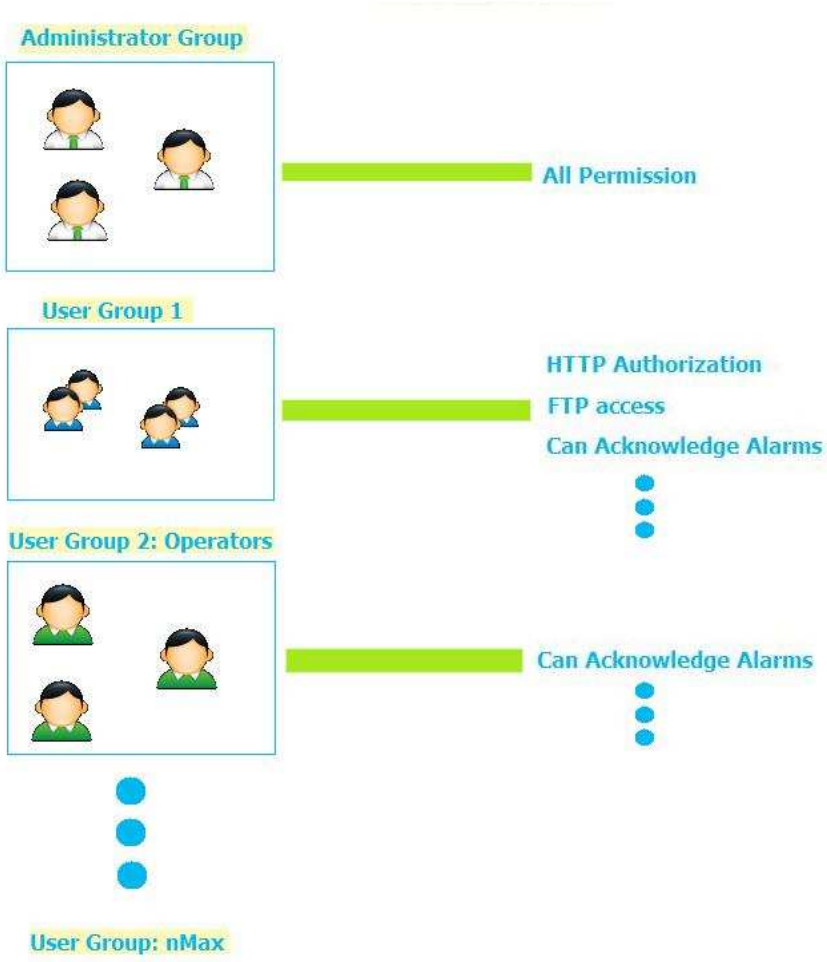


Figure 193

22.1 Configuring Security Options

The section describes how to configure security settings in the JMobile Studio.

NOTE *To enable/disable the user management feature, right click on the "Security" folder in the Project View and set Enable or Disable. See the following figure as a reference.*

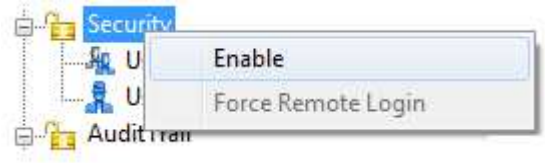


Figure 194

22.2 Configuring Groups and Authorizations

Open **UserGroups** to configure in the **ProjectView**.

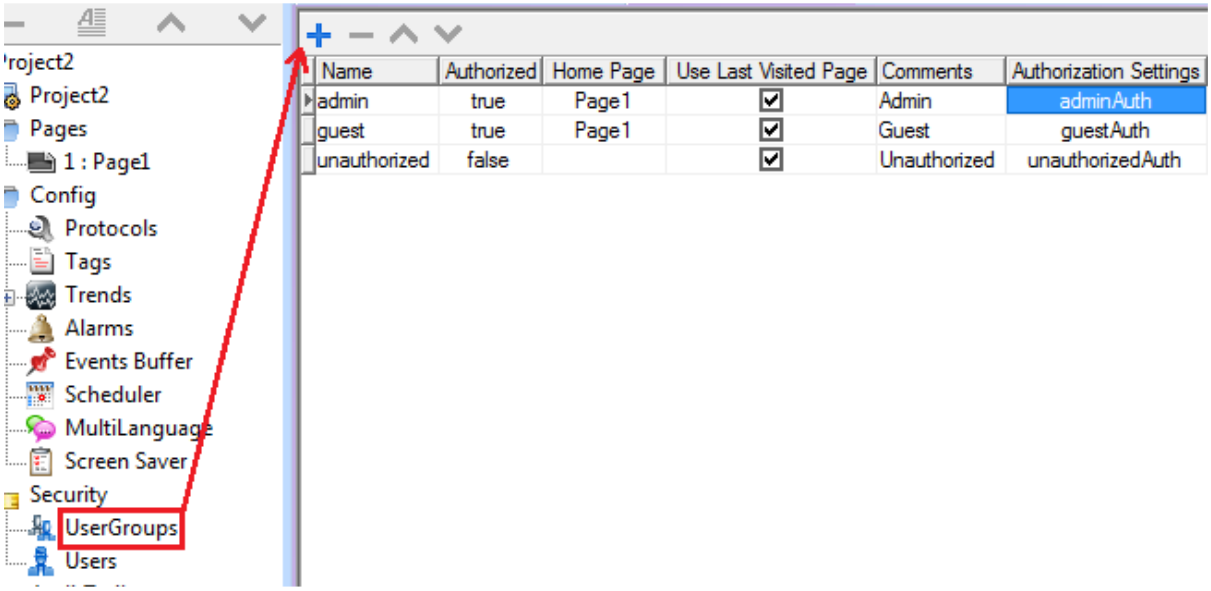


Figure 195

New User groups can be added by clicking the "+" Button.
Three predefined groups are available by default, these groups cannot be deleted and their names cannot be changed.
Predefined group authorizations and comment fields can instead be changed according to the application's requirements.
For each group of users you can assign a Home Page. This means that, whenever a user from this User Group is logging in, the selected Home Page for that group will appear.
There is one additional option called 'Use Last Visited Page'. If enabled, and a user logs in, the page visited by the previous user will be displayed.

22.3 Modifying the Access Permission of Groups

To modify and assign the permissions, click the browse button on the Authorization Setting column.

The Admin Authorizations dialog will open, giving you access to tabs for the different available options.

22.3.1 Widget Permissions

The following figure shows the dialog where you can change the widgets permissions.

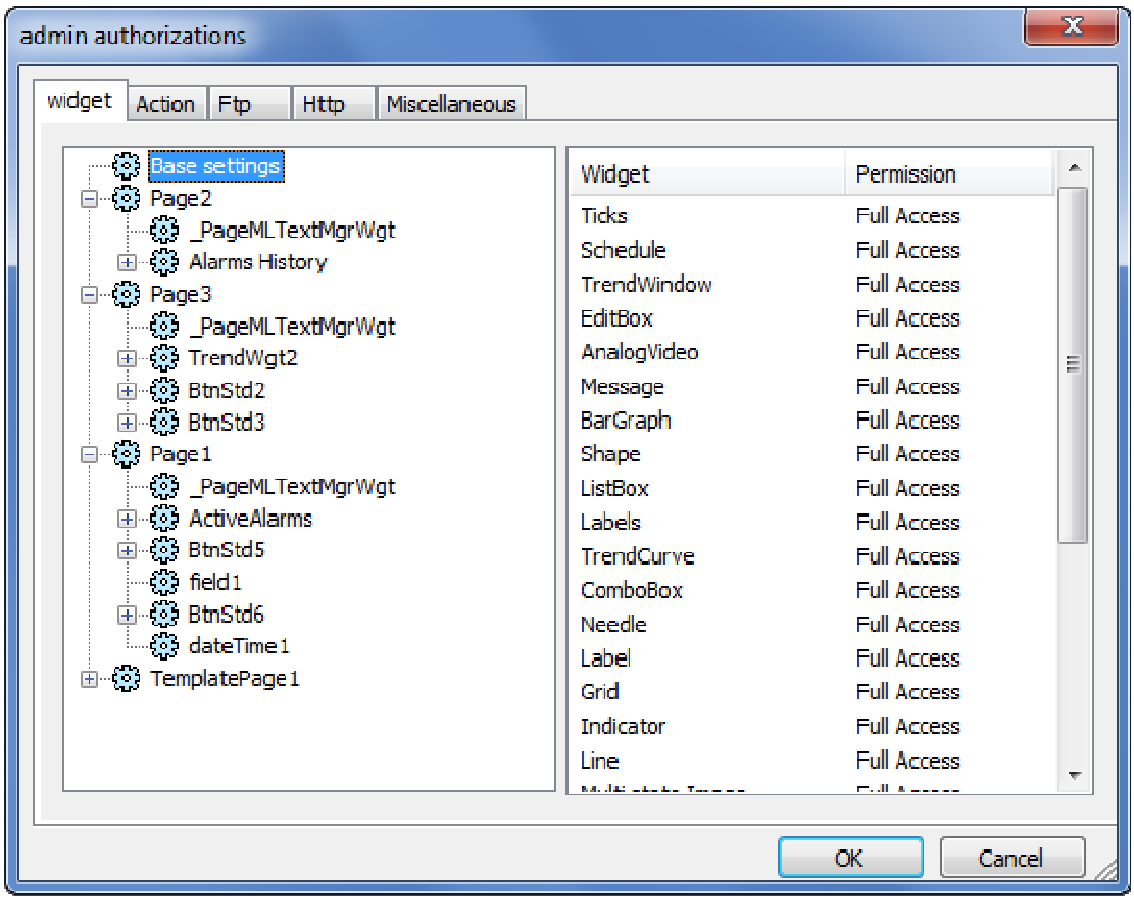


Figure 196

For the widget, the possible options are:

- Full-Access,
- Read-Only
- Hide

When you click on “Base settings” the right part of the dialog shows the permissions that will be valid as default and at the project level.

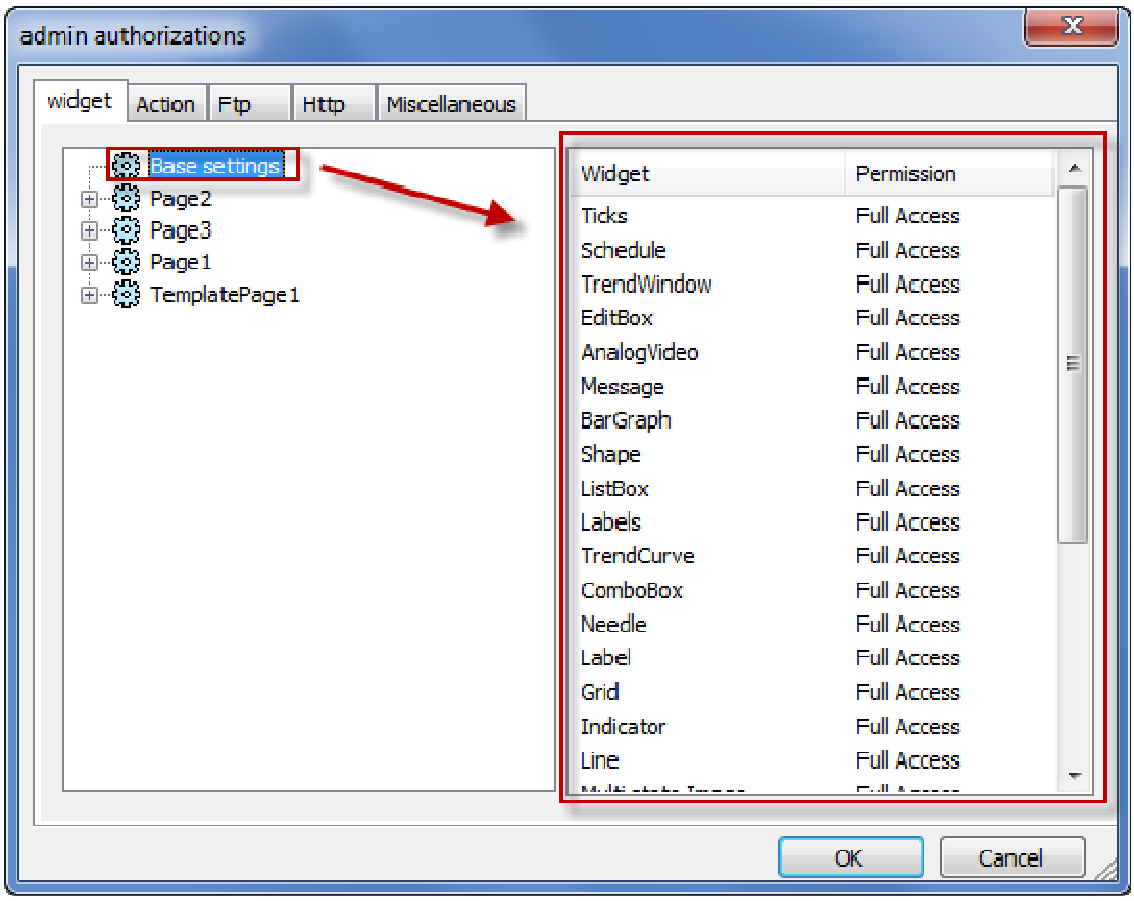


Figure 197

The widgets' security settings can be changed, not only globally, but also for each single widget defined within the project; all the widgets can be reached from the tree structure on the left part of the widget tab.

Permissions can be given at three levels:

- Project level
- Page level
- Widget level

In the tree structure the permission for a page can be set as

- Full Access
- Hide
- Read Only

All the widgets on this page will take the settings that have been assigned to the page with a type of hierarchy logic.
Suppose the page permission is set as 'Read Only', then all the widgets in the page will have the permission as "Read Only". On selecting a widget inside the page from the tree structure, you can see that the permission is given as "Use Base Settings". This means that it takes the permission given to the page (Read Only).

The widget permission takes the priority as follows:

- Low priority — Basic settings (widget settings in general for the project)
- Medium priority — Page settings (settings for all the widgets on a particular page)
- High priority — Widget settings (individual widgets or its group/parent widget permission of any page).

For example, suppose a widget is set at “Read Only” permission at project Level and it is given “Full Access” at page Level then the page Level Settings will be taken.

Later in the chapter, we explain how to modify permissions for a specific widget directly from the page view (rather than locating the widget from the tree view shown in the authorization dialog).

22.3.2 Action Permissions

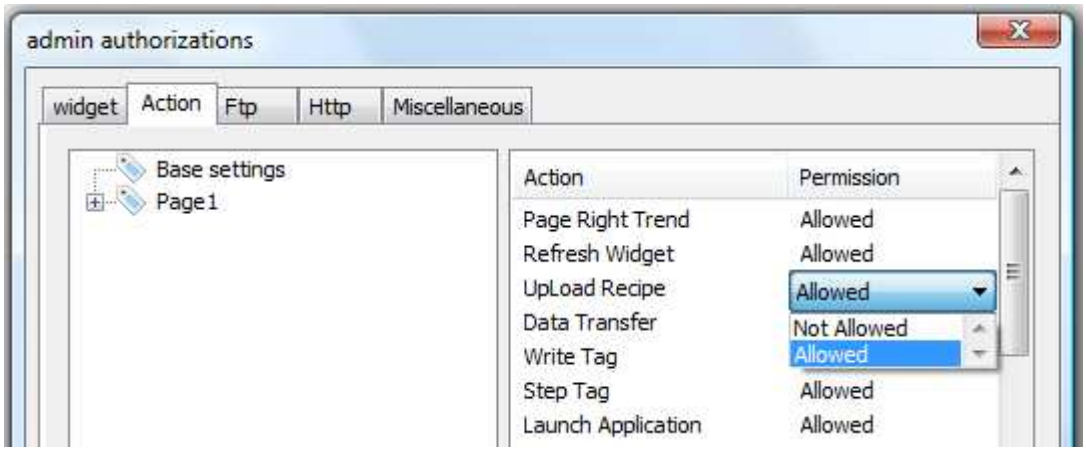


Figure 198

With this dialog, it is possible to assign the authorizations for the actions with respect to a project. The access is either Allowed or Not Allowed.
As for the widgets, the authorizations can be assigned globally, but also for each single page, and the widget programmed into the project.
Later in the chapter, we will explain how to modify permissions for a specific action directly from the page view (rather than locating the action from the tree view shown in the authorization dialog).

22.3.3 FTP Authorizations

For each group of users you can set specific authorizations related to the use of the FTP server.

FTP permissions can be enabled or disabled. If enabled, you can specify from the "Permissions" combo box the access level selecting between All, Write, Read, Browse, and None.
The IP Address list access allows you to specify from which IP an incoming FTP connection should be accepted.

NOTE *IP access list configuration is common to all groups.*

22.3.4 HTTP Authorizations

The HTTP authorization dialog allows to configure restrictions related to http access to the web server integrated into the runtime. HTTP settings are common to all groups and are valid just if security flag is enabled.

IP list can be used to list allowed ip addresses. Default is **Allow all.** Only IP listed in **IP list** will be authorized to access to http server embedded into the runtime.

Access limits is used to allow or restrict access to particular files and folders into the workspace. Based on **Force Remote Login** flag default workspace access change and as conseguence using **Access limits** is possible to open or close access to specific resources.

Force Remote Login	Default Access to workspace	Access limits
-	FULL	-
Disable	FULL	Can be used to block access to some files/folders or to require auth for it
Enable	No Access	Can be used to open access to files/folders

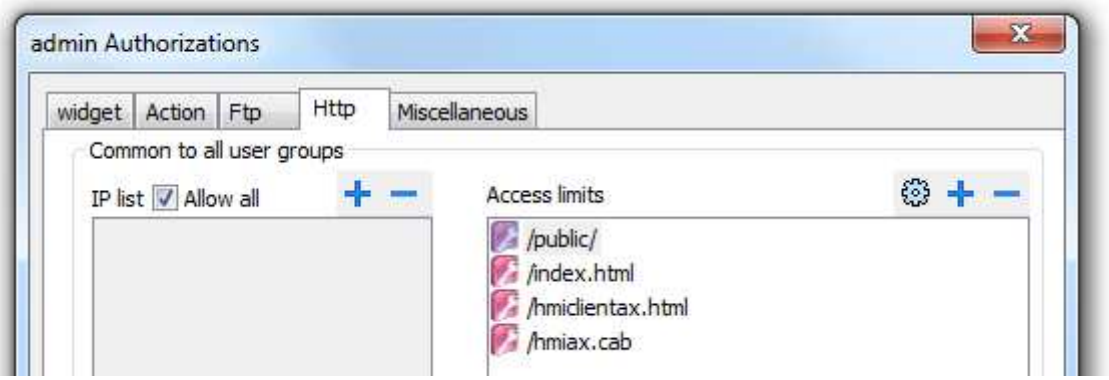


Figure 199

Set default access limits icon on the left of **"+ - "** can be used to restore default configuration removing user customizations. Default is allow following public resources:

- **PUBLIC** folder and Index.html, that contain web console and public resources
- **ActiveX** files (hmiclientax.html, hmiax.cab)

22.3.5 Miscellaneous

The Miscellaneous tab contains different settings related to several options as indicated in the following picture.

Please note that as indicated in the picture, some settings are related to the group, but some settings are global to all groups.

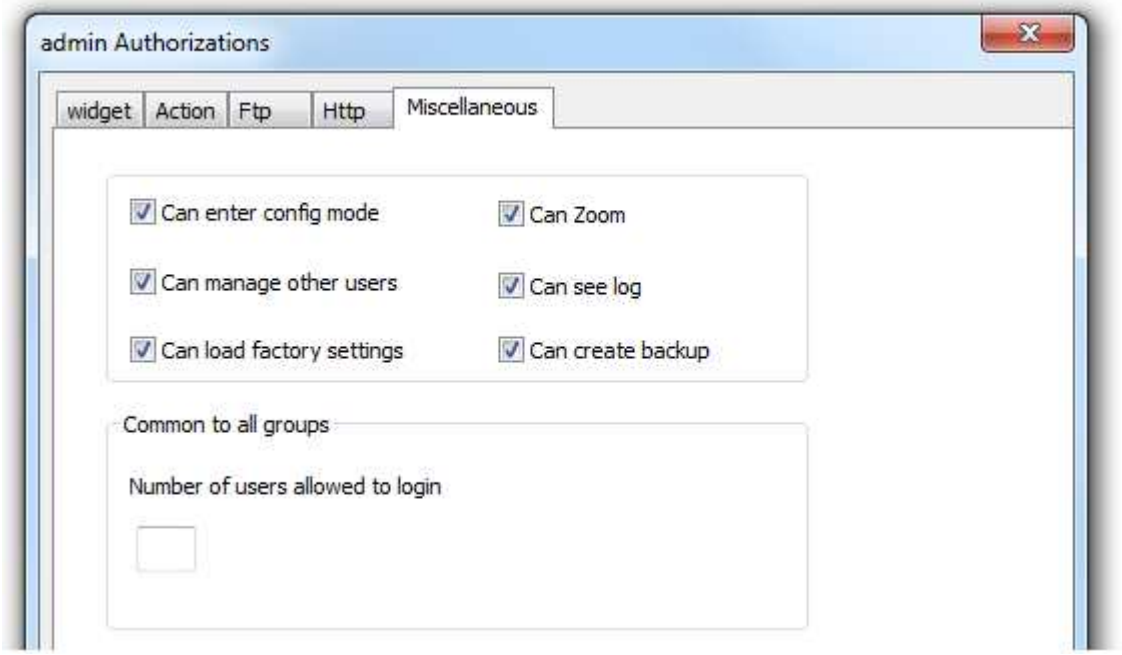


Figure 200

Can enter config mode	Allow users of group to move runtime to configuration mode (for maintenance usually).
Can manage other users	Allow users of group to manage other users like a superuser at runtime. A user with this permission can add new users, remove users or change user permissions.
Can load factory settings	Users setting can be changed at runtime by authorized users and are saved into internal storage usually. A user with this authorization can execute a macro to clear these dynamic files and restore user management setting as was at beginning after first project download.
Can zoom	Allow user to zoom in/out using context menu at runtime
Can see log	Allow user to see logs at runtime
Can create backup	Allow user to backup project.
Number of users allowed to login	Max number of users that can be connected to runtime in the same time. Default is 3.

22.3.6 Access Priority

If the Access control is applied to a Widget, page and or even the Global Access, then the top priority goes to the Widget access.

- Top Priority — Control from Widget
- Medium Priority — Page Access or its Parent Access
- Low Priority — Global Access

This means that “exceptions" configured for an action or a Widget, directly from the page view, has priority over the base settings.

22.4 Configuring Users

To configure users double click on Users from the Project View, and then click on the + sign to add a new user. A user named **admin** is already present by default and this user cannot be deleted.

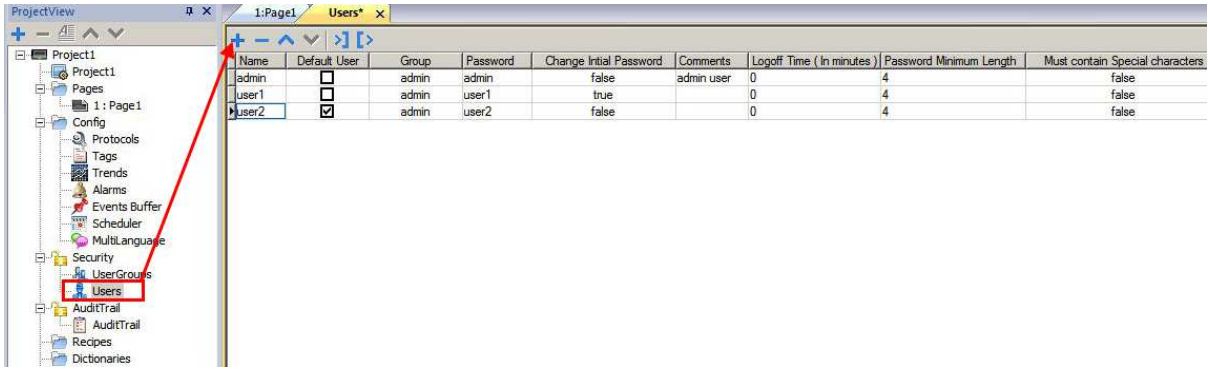


Figure 201

Name	User Name
Default User	Identifies the user which is automatically logged-in by the system when starting, re-starting or after a logout; only one default user is allowed.
Group	Groups of the user. Groups are used to assign authorizations to users.
Password	Password for the user
Change Initial Password	If True, the user is forced to change his password on first logon
Comments	Comments for the user
Logoff time (In Min)	The user will be automatically logged off after the specified time with no actions on the panel. After Log off, the Runtime goes to the default user.
Minimum Length	The minimum length of the password. It should be equal or greater than the set value.
Must Contain Special Characters	If True, the password should contain at least one special character
Must Contain Numbers	If True, the password should contain at least one numeric digit.

22.5 Default User

You can program a Default User for a project. When the system starts or reboots, the Runtime is logged in with the default user. All the privilege settings of the default user will be activated in the system. If you want to log in as a different user in Runtime, you can use either the Switch User macro or the Log Off macro.

The default user will automatically get logged in if any user (other than default user) logs off.

22.6 Assigning Widget Permissions from Page View

You can assign different levels of security, to different user groups, on a single widget, directly from the project pages.
Select the widget, then right click and select security settings from the context menu. Next, choose the group and assign the security properties to access the widget (as shown in the figure).

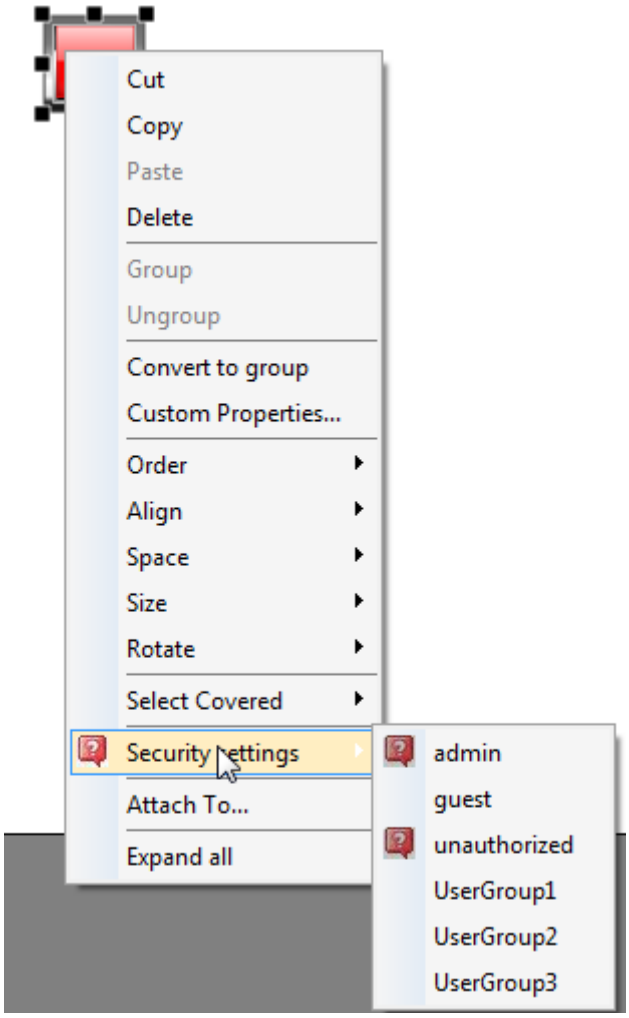


Figure 202

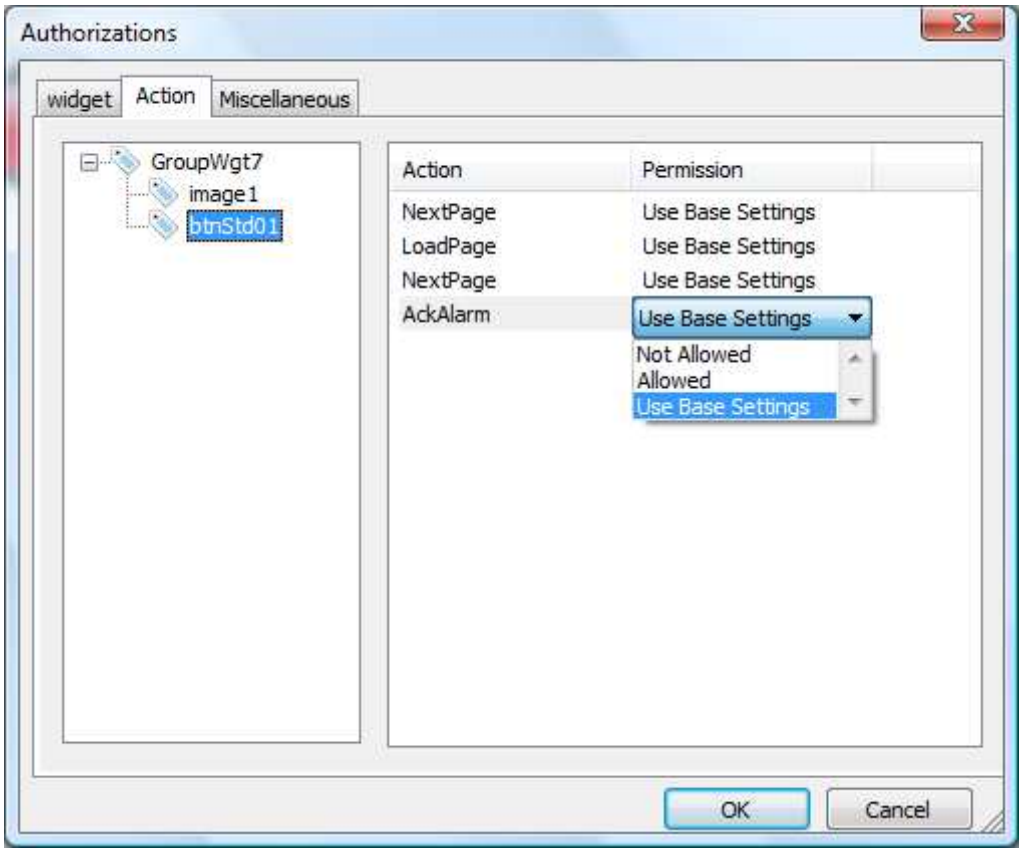


Figure 203

22.7 Operation on Runtime

After starting the Runtime, if **a default user** is specified within the project, the system will provide automatic login of that user without prompting for a user login. If no default user is configured, the system will ask for a User name and Password, and based on the user, the Runtime will allow only the configured permissions for that logged user.

There are specific actions for user **logout, edit user**, **add user**, **remove user** and **switch user**.
Users can be edited, added or removed on Runtime as explained in the chapter User Management Actions.

All the users' information modified at Runtime is stored in a separate file, thereby preventing loss of the users' configurations in case of a new project download. To remove dynamic files and changes applied to user's configuration during runtime there're two ways:

- Runtime side: **DeleteUMDynamicFile** action
- JMobile Studio side: **Delete Dynamic Files** flag available in download dialog.

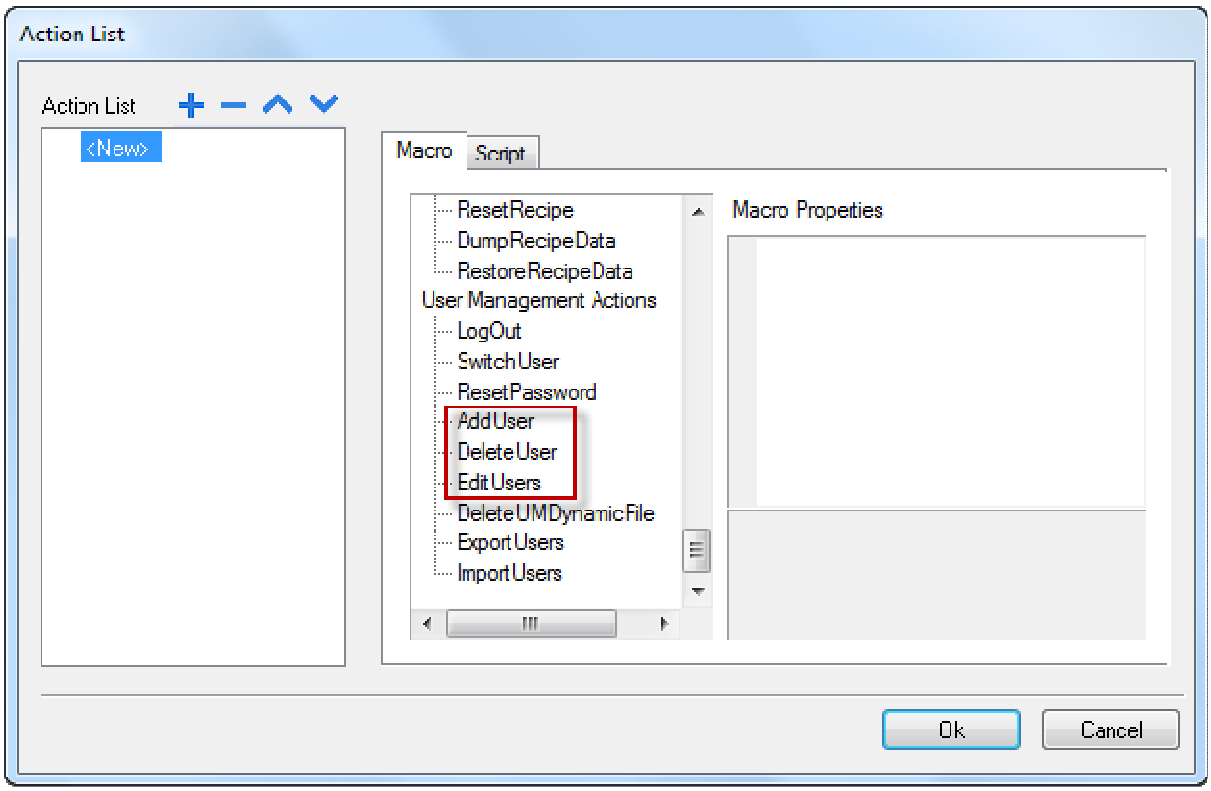


Figure 204

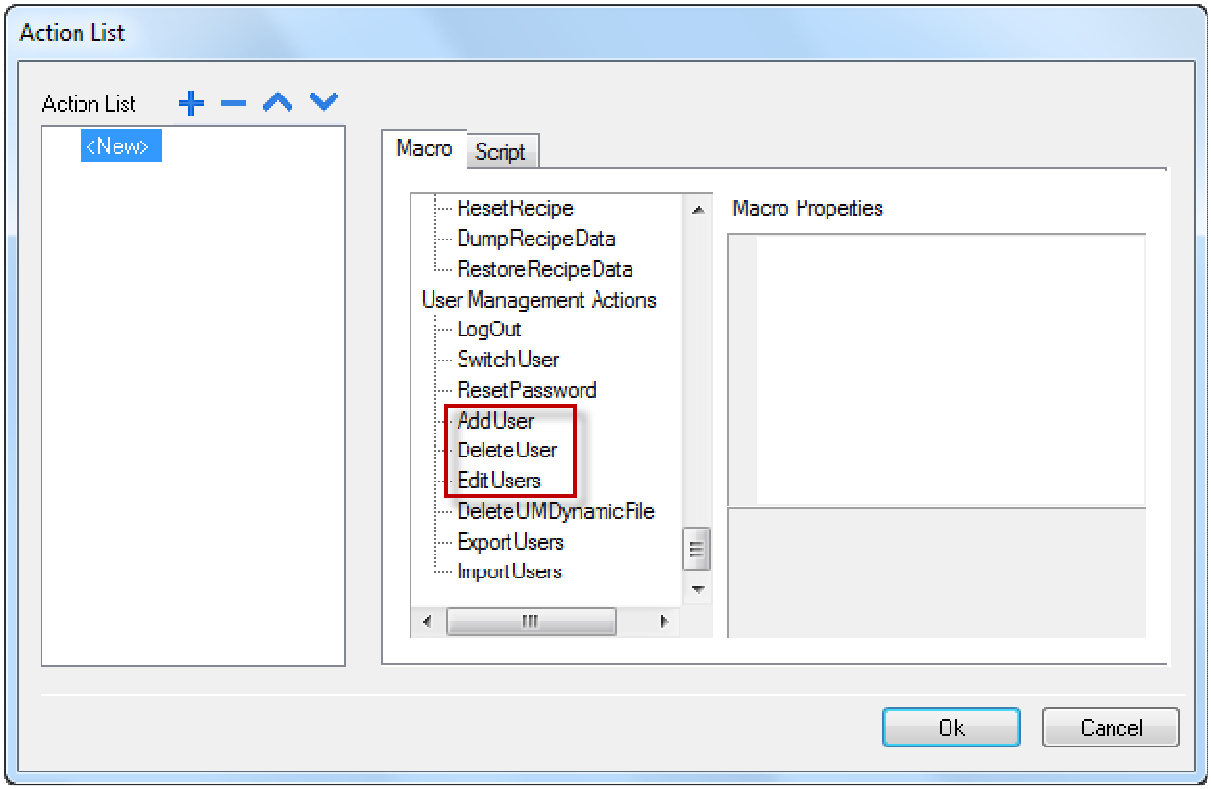


Figure 205

22.8 Force Remote Login

Starting from v1.9 of JMobile Studio, a new flag is available to force user to LogIn when using remote access (via Activex or JMobile Client), this is working when user management is enabled. If **Force Remote Login** is not enable remote access will use same level of protection of local access.

Force Remote Login is useful in particular when a default user is configured in runtime to automatically login without having to enter a login and password at startup, but a remote access protection is required.

Figure 206

Force Remote Login, when enabled, blocks all access from the web to the workspace folder in runtime. The only files/folders still accessible when this flag is enabled are by default:

- **PUBLIC** folder and **Index.html**, that contain web console and public resources
- **ActiveX** files (**hmiclientax.html**, **hmiax.cab**)

Please check **Security** -> **UserGroups** -> **Authorization Settings** -> **HTTP** tab for more details related to HTTP access limits or chapter **HTTP Authorizations.**

23 Audit Trails

JMobile Studio supports Audit Trail functionality which provides basic process tracking and user identification linked to events with a time and date stamp. The logged users and events allow for a review and/or report of your production processes.
The Audit Trail function provides flexible, tailor-made and easy-to-review event logs.

The Audit Trail (or audit log) is a chronological sequence of audit records, each containing information on the actions executed and the user that did them.
The Audit Trail can be enabled with or without user management. So it can access and supervise all actions from all users, and a normal user could not stop or change this.

23.1 Enable or Disable the Audit Trail

In the Project View pane, right click on the Audit Trail and click either enable or disable the Audit Trail recording on Runtime. The padlock symbol in the tree informs you that, in the project, the Audit Trail is enabled or disabled. When the Audit Trail is enabled, the padlock symbol is shown locked, otherwise, it stays open.

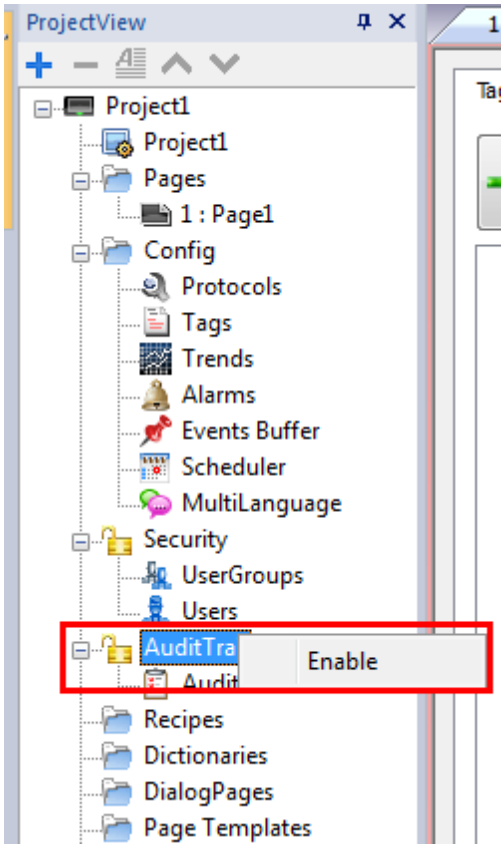


Figure 207

23.2 Configure Audit Events

You can have more than one set of Audit Records. To add to the Audit files, you need to configure the Events buffer.
Double click the Events buffer from the project workspace. Next, add the events buffer and set the file size, and then select the log type “Audit".
Here there is an option for selecting the storage where the dumped Audit files have to be stored.

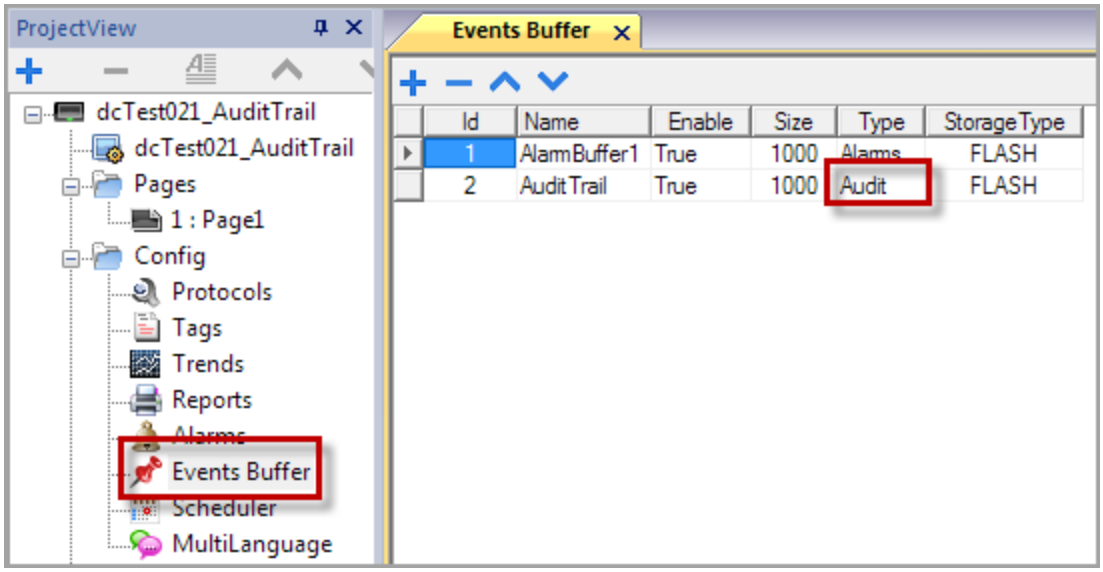


Figure 208

The system provides a save to file on the disk every 5 minutes.

23.3 Configure Tags in the Audit Trail

For most cases, all the tags specified in the project do not necessarily need to be monitored. You can customize the tags to be monitored by the Audit Trail.

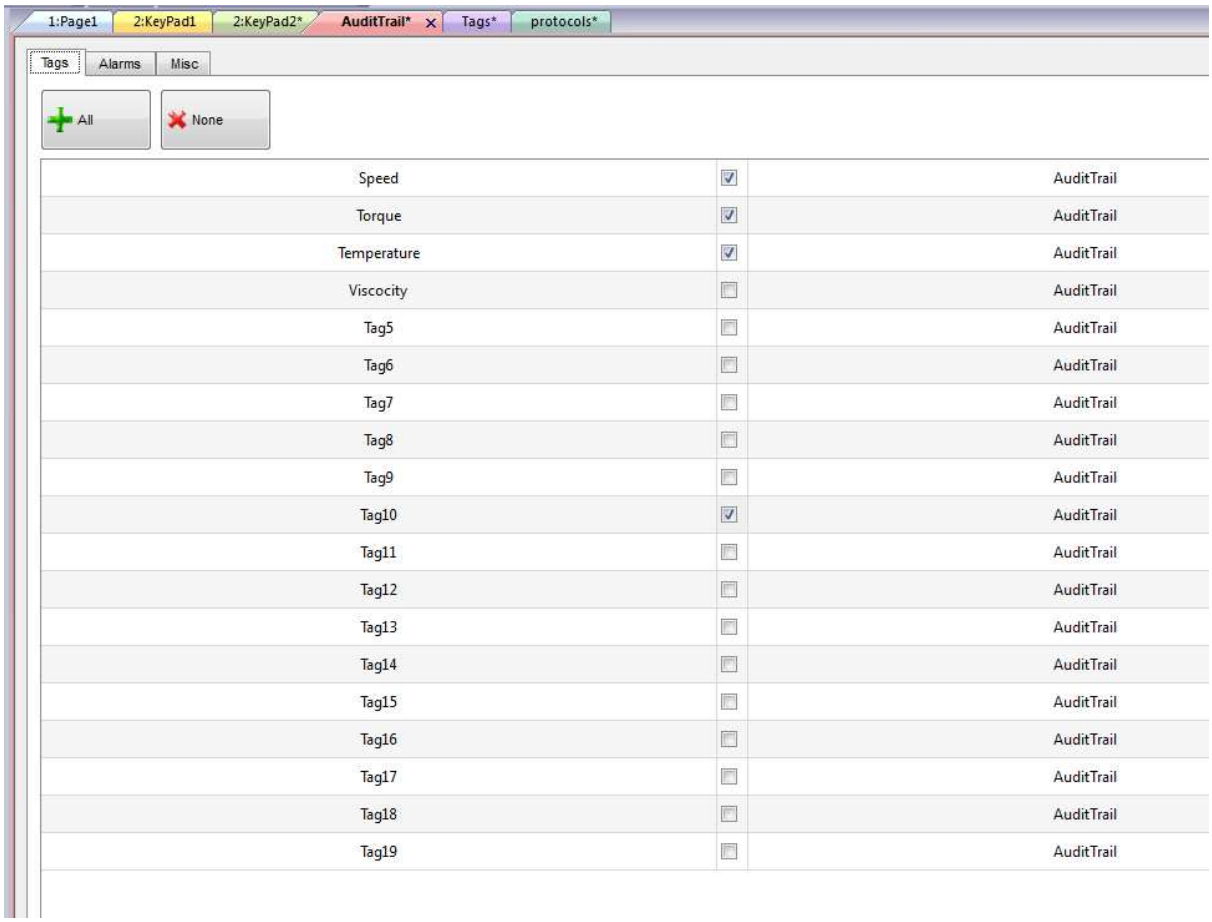


Figure 209

In the Audit Trail editor, all the Tags are available for selection. You can select only the Tags to be monitored by the Audit Trail. For each selected Tag, the Audit Trail will record the write operation to that Tag, together with the time stamp and user that executed the write operation.

23.4 Configure Alarms in the Audit Trail

You can specify the alarms to be monitored by the Audit Trail. Double click Audit Trail from the project workspace and click on the Alarms tab. Select the alarms you want to be logged in the Audit Trail. The Audit Trail for alarms will also record and acknowledge the operation done by the logged-in user.

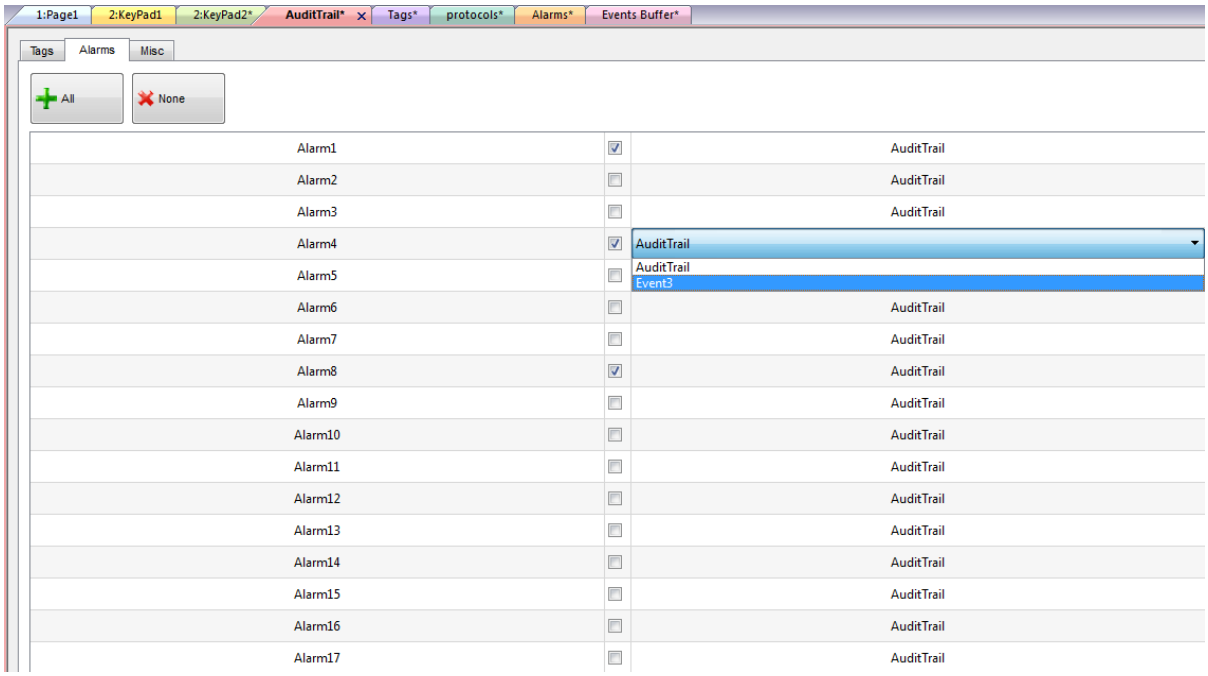


Figure 210

23.5 Configure Login or Logout Details in Audit Trail.

The Audit Trail can record information about user login and user logout events. These settings are available in the Misc tab of the Audit Trail.

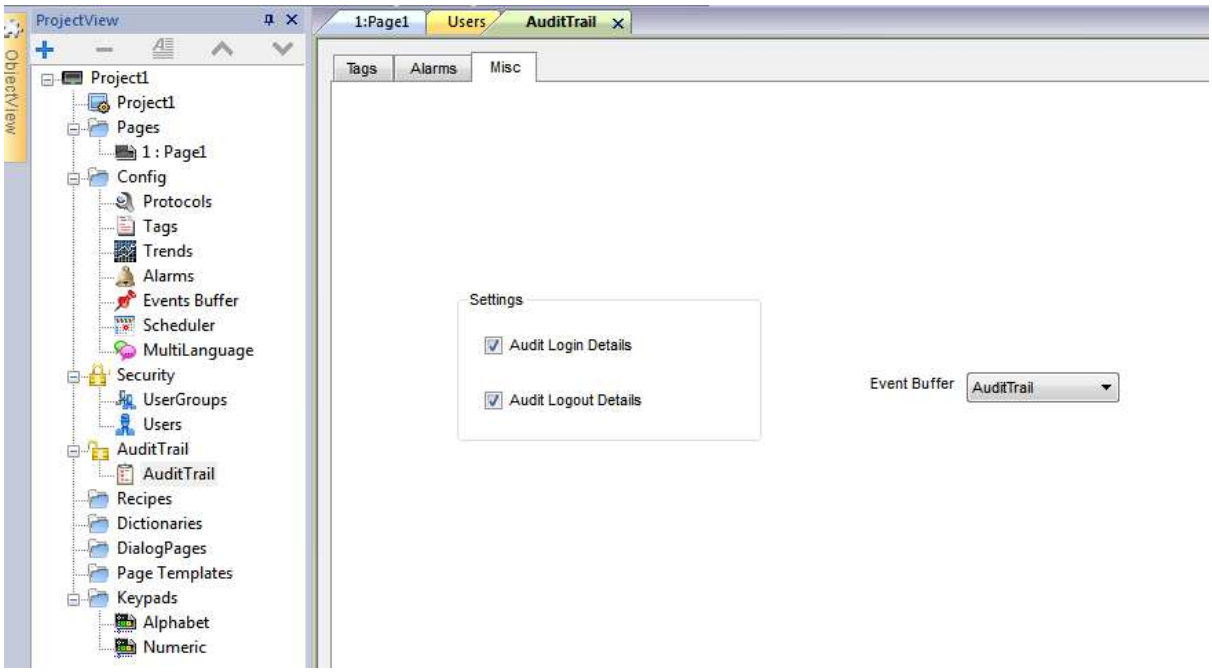


Figure 211

23.6 Viewing Audit Trails in Runtime

The Audit Trail data cannot be displayed in Runtime. It is only available in the exported data file.

23.7 Exporting Audit Trail as CSV File

You can convert the audit data to a “.csv" file.
For a detailed description, look at the explanation provided for the DumpArchive macro action.

24 Reports

A report is a collection of information that will be printed when triggered by an event.
The JMobile Studio programming software provides an editor to configure reports, their content, the printer and the trigger conditions.
The report comes as a special collection of pages with header, footer and body, including options for cover page. When configuring reports, JMobile Studio provides access to a dedicated widget gallery featuring only the widgets available for reports.
When the programmed event is triggered, the report printout is started and the entire printing activity is carried out in the background.

NOTE *Printing of reports is not supported in remote (using for example JMobile Client or ActiveX).*

24.1 Adding a report

In the Project Workspace, double click on **Reports** to open the Editor. Then add the report by clicking the “+" button.

Two types of reports are available:

- **Text Report**
- **Graphic Report**

Text Reports are used to configure line-by-line printing of alarms. Text Reports are designed to work with line printers. Text is sent directly to printer’s port without using any special driver. Not all printers support this operation mode. This printing mode only works in WinCE platforms and requires to use a physical port.

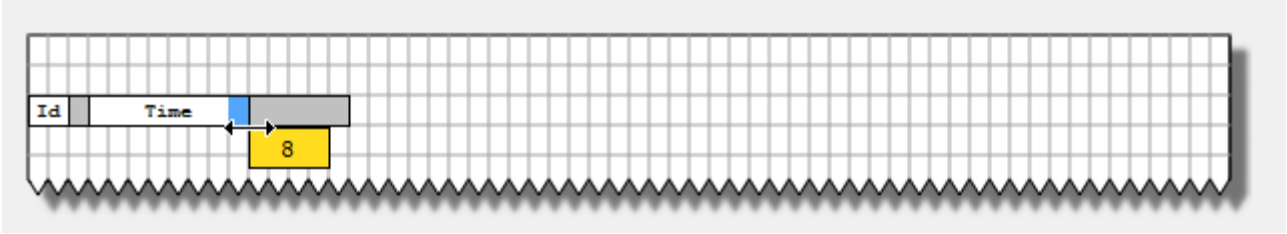
Graphic Reports contain graphical elements and may include complex widgets such as screenshots, or alarms. A specific printer driver for each printer is required for printing graphic reports; the list of supported drivers is in following chapters.

24.2 Text Report

To add a Text Report for line-by-line alarm printing click on the *Text Report* button in *Reports* toolbar.

The format of the report can be freely defined using the report editor; the paper size can be defined in number of characters, while the available fields are listed in the box on the right side.

To include a field in the line to be printed, just drag and drop it from the list to the page layout.
The field can be resized using the mouse; the tooltip shows the dimension in “chars”.



In case the text cannot fit in the dedicated space, than the auto wrap is applied.

Printer options can be used to control flush of pages in printer. Depending on the printer, text can be printed immediately or after a timeout (from few seconds to minutes). However, it is always possible to force flush when one of following conditions happens: after n events, after n lines or after n seconds. A temporary buffer is used by runtime software. Flush conditions are in OR, so, as soon as a condition is met, the page will be flushed out of printer.

NOTE *Not all printers support Text Report operation mode.*

NOTE *Text Reports only work in WinCE platforms and require to specify a physical port (PDF format is not supported by Text Report.)*

NOTE In *line printing, text is printed immediately line-by-line or after a timeout. This timeout may depend on printer model (could also take minutes for some models not designed for line printing).*

24.3 Graphic Report

To add a Graphic Report, click on the *Graphic Report* button in *Reports* toolbar.

The following figure shows report configuration editor.

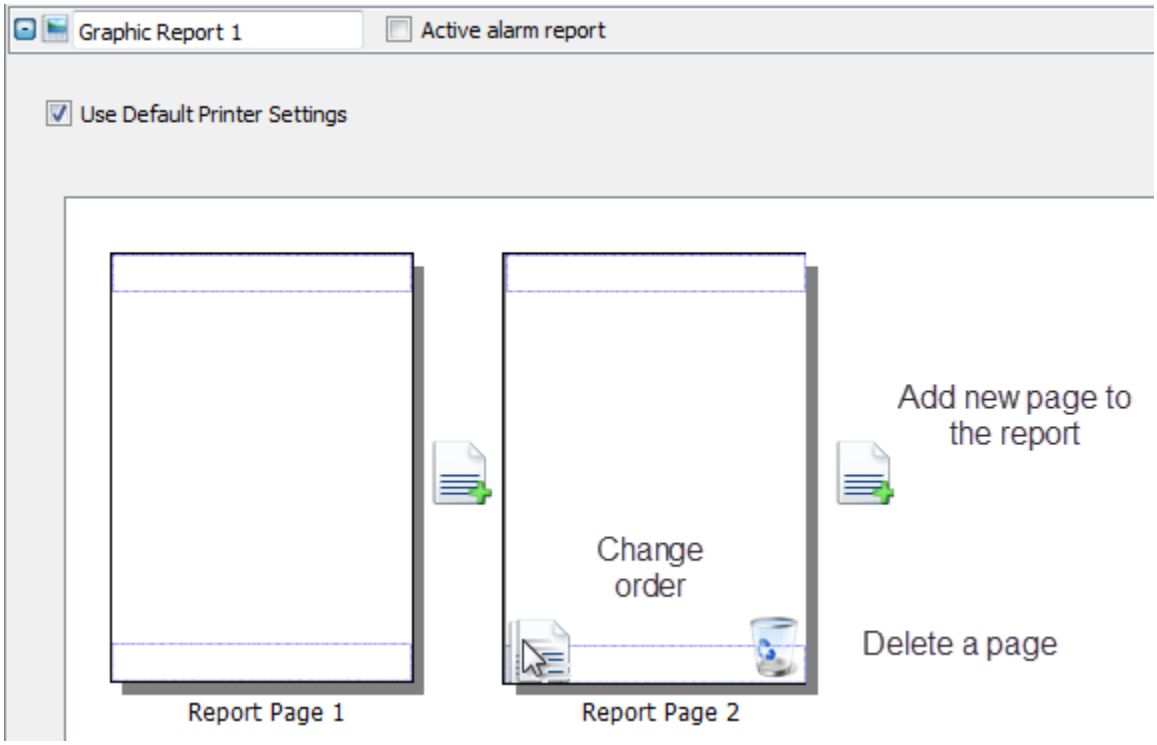


Figure 212

This part of the editor is used to set the number of pages and their order.
Use the icon with the “+” symbol to add a new page to the report layout.
When the mouse goes over a page already configured, two icons appear to allow reordering or deleting pages.

Double click on a page to edit the page report content using the page editor.
Each page is divided in three sections: the header, the footer and the page body.
In the page editor the area under editing is shown in white, the others are grayed out.
To edit a different section, just double click over the grayed out area.

24.3.1 Page body

The page body is the central part of the page.
The widget gallery accessible from the right side sliding tab is context-sensitive and includes only the widgets available for the area under editing.

24.3.2 Header and Footer

Header and footer are respectively the top and bottom parts of the page.
The widget gallery accessible from the right side sliding tab is context-sensitive and includes only the widgets available for the area under editing.

24.3.3 The Context Widget Gallery

The widget gallery which can be normally recalled from the right side sliding pane is always adapting itself to the context.

The available widgets are:

Page Number Widget
Automatic page numbering

Screenshot Widget
Used to take a print screen of the current page HMI is showing. When you drag & drop the widget on the page it will get automatically get the page dimensions of the HMI.

NOTE *The system will print the full area of the screen. So, all dialogs opened will be printed as part of this area.*

Alarm widget
Used to print the entire contents of the event buffer (the Default buffer is Alarm Buffer1)

The "**Text**" category collects the typical widgets used to compose reports with labels and numeric fields.

24.3.4 Printer Configuration

A default printer can be configured from Printer Setting menu for all graphic reports. Each report can be configured to use it or to use a different type of printer.

For PDF printer (supported only by Graphic Reports), you can to define the folder where files are saved by using **Printed Files Location**.

24.3.5 Supported Printers

The table shows the list of print languages supported by Windows CE driver **printCE.dll** (driver in use in Windows CE platform):

HP PCL 3, HP PCL 5e, HP PCL3GUI	printers compatible with **HP PCL3/PCL5e/PCL3GUI**, including models many **DeskJet**, **LaserJet, DesignJet**
Epson ESC/P2	printers compatible with **ESC/P2**, **LQ**

Epson Stylus Color	printers compatible with **Epson Stylus Color**
Epson LX (9-pin)	9-pin printers compatible with **Epson LX, FX, PocketJet**
Cannon iP100, iP90, BubbleJet	printers compatible with **BubbleJet, iP90, iP100**
PocketJet II, 200, 3	printers compatible with **Pocket Jet**
MTE Mobile Pro Spectrum	printers compatible with **MTE Mobile Pro Spectrum**
Adobe PDF File	**Adobe PDF** file
SPT-8	printers compatible with **SPT-8**
M1POS	printers compatible with **M1POS**
MP300	printers compatible with **MP300**
Zebra	printers compatible with **Zebra** CPCL language.
Intermec PB42, PB50, PB51, PB2, PB3	printers compatible with Intermect **PB42/50/51/2/3** with ESC/P language
Datamax Apex	printers compatible with **Datamax Apex**

Supported ports:

- LPT1 (USB printers)
- File (PDF)

NOTE In Win32 platform, the only supported printers are **PDF** and **Default.** Default is used to indicate default OS printer configured in target. Any printer (not only USB printers) can be used in Win32 platform.

24.3.6 Printer tested

Follow the list of printers tested with printCE drivers in WCE targets.

Driver	Printer Model	Graphic	Line
Epson ESC/P 2	Epson AcuLaser M2310	Yes	Simulate
Epson LX (9-pin)	Epson LX-300+II	No	Yes
HP PCL 3	HP LaserJet P2015dm HP LaserJet 4700dtn	Yes Yes	Simulate Yes

HP PCL 3 GUI	HP Deskjet 1010 HP Deskjet D5560 HP LaserJet 4700dtn	Yes Yes No	No No Yes
HP PCL 5e	HP LaserJet P2015dm HP LaserJet 4700dtn	Yes	Simulate
INTERMEC	Intermec PB50 with ESC/P language with 4 inch roll paper. Note: HMI crash when trying to print on Intermec PB50 printer in standby mode after a first successful print.	Yes	Yes
PDF		Yes	No

24.4 Print Events

The configured reports can be triggered by specific events.
For Alarms, the configuration of the events can be done directly in the alarm editor from the Events dialog by clicking on the **Print** tab as shown in figure.

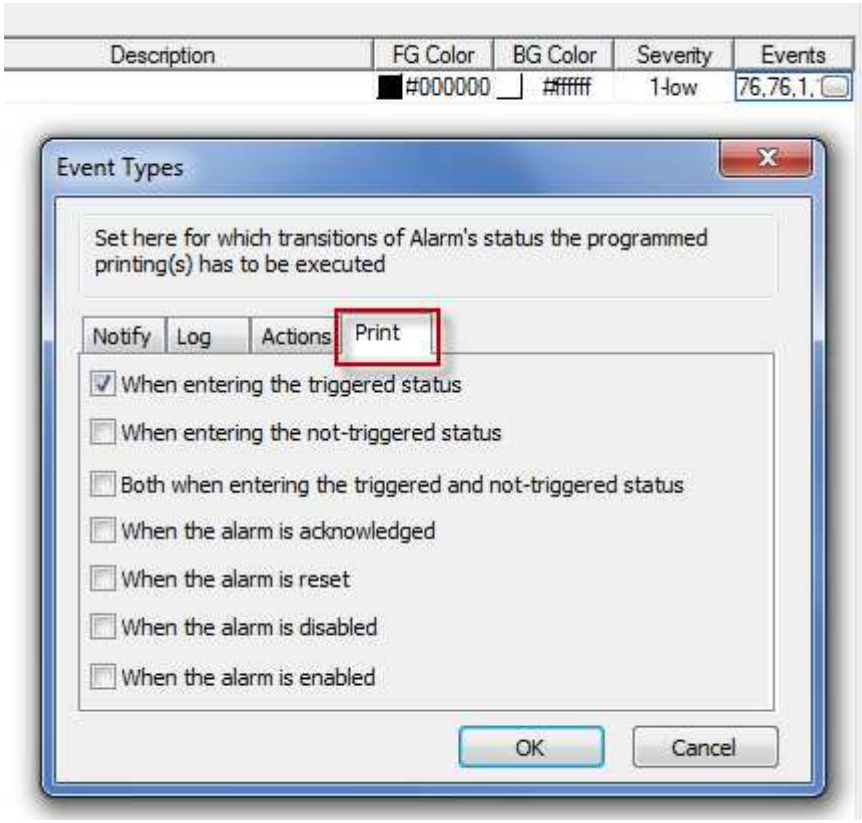


Figure 213

Only one report can be set as **Active alarm report** in a project. An alarm report can be a *Text Report* or a *Graphic Report*.

A Graphic report printing can be started also using the dedicated action call **PrintGraphicReport**.

The **Silent** option (**true** by default in action settings) allows, when set to false, a dialog to pop-up at runtime asking the user to adjust printer settings as shown in figure.

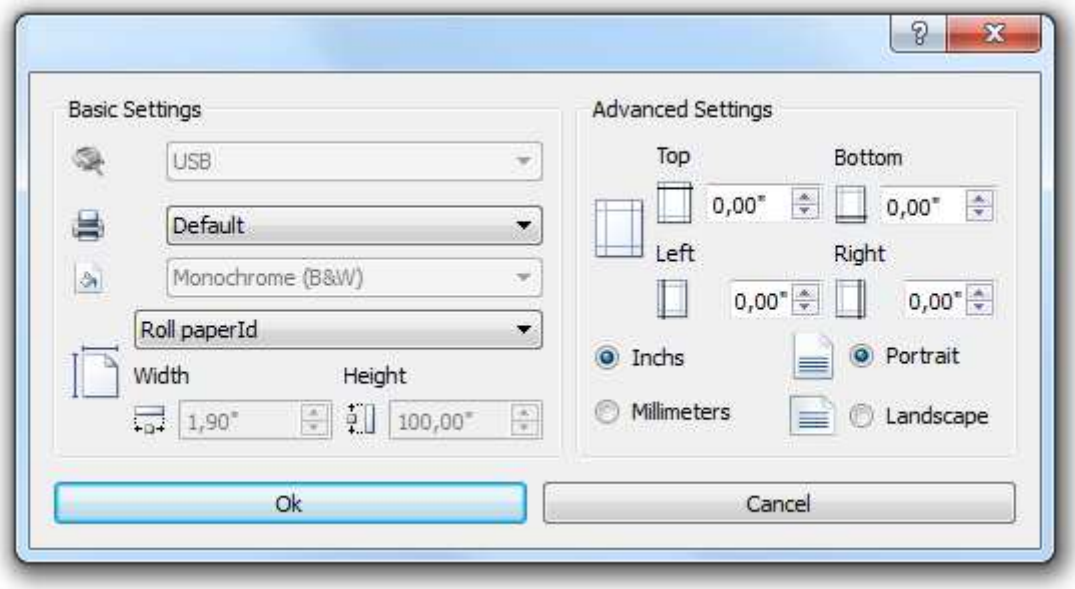


Figure 214

24.5 Minimum requirements

Report printing requires operating system (BSP) V1.54 or above for Windows CE devices.

25 Screen saver

Screen saver can be used to show a slideshow when the HMI is not in use. Screen saver start when one of following events does not happen for a certain time range (**Timeout):**

- **Touch of display**
- **Mouse move**
- **External keyboard key pressed**

Screen saver configuration is available in JMobile Studio in **Config -> Screen Saver** section.

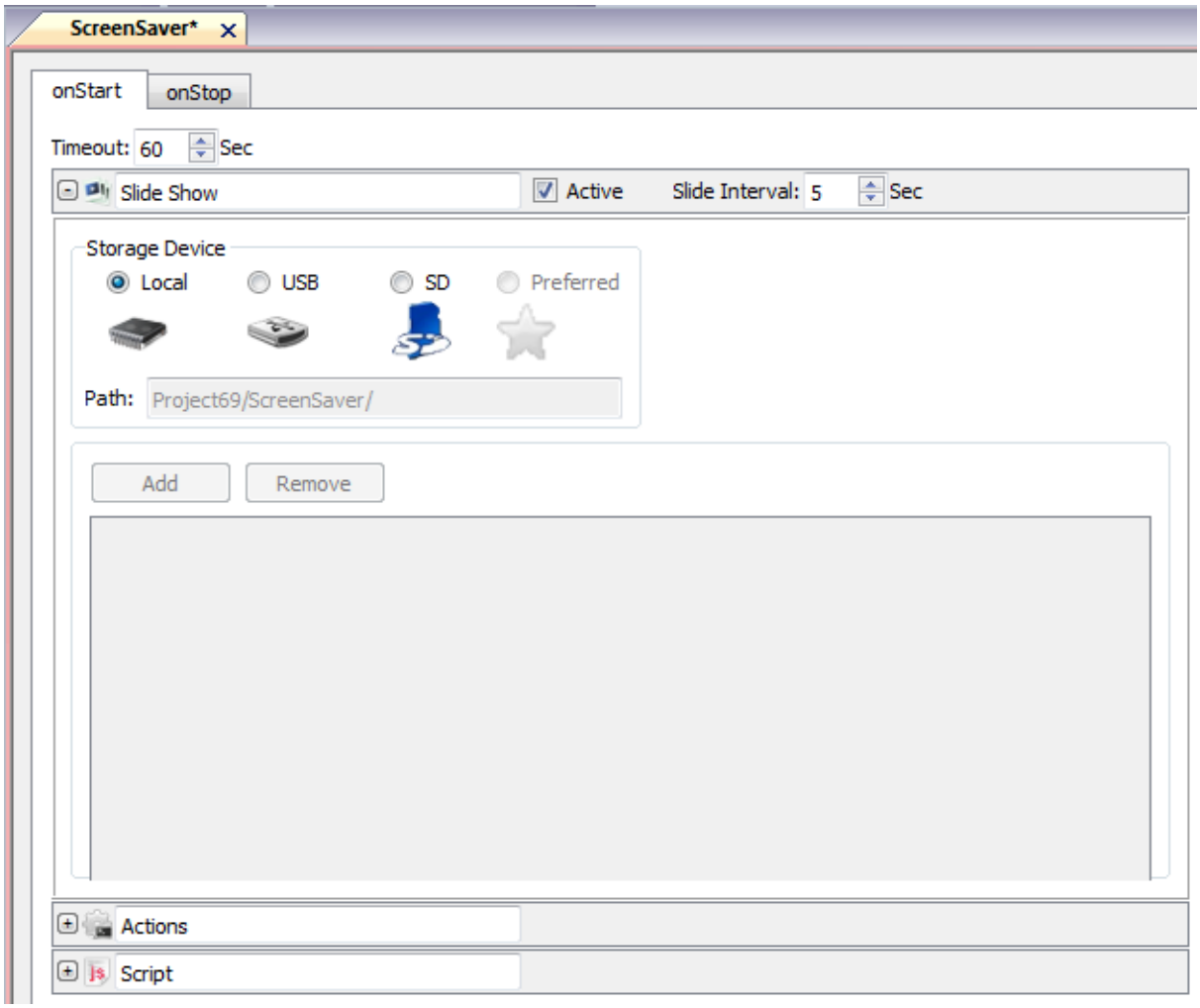


Figure 215

To configure screen saver as slideshow, proceed as follow:

1. Right click on **Screen Saver** from the project tree and click **Enable**
2. Select **Timeout** value (number of seconds before screensaver start when there's no user interaction)
3. Select **Slide Interval** (the number of seconds before switch slide)
4. Select **Storage Device** used for reading images used by slide show (Internal Storage, USB or SD).

For internal storage (Local), it is possible to select and import images that later will be downloaded into the device at project download. Images are downloaded into the folder *workspace\projectname\screensaver.*

When an external storage is used, images are located in the folder *screensaver* available in USB or SD devices.

The supported image formats are: JPEG/PNG.

When the screensaver starts/stops, it is possible to execute some actions (macros or JavaScript functions) In Tab **onStart** actions can be configured to execute when the screensaver start, in Tab **onStop** actions will execute when the screensaver stops.

The Screen saver is supported by WCE & Win32 runtime. Is possible to use screensaver also in JMobile Client & ActiveX clients.

26 Backup/Restore

Backup/Restore of the HMI Runtime and project is available.
Backup operation is working as follow:

1. Automatically unload current project to unlock opened files in use
2. Archive in a .zip file (standard or encrypted) the content of qthmi folder that contain runtime, projects, dynamic files like recipes / alarms / trends etc.
3. Reload project

Backup can be executed from the context menu in runtime -> **Backup.**

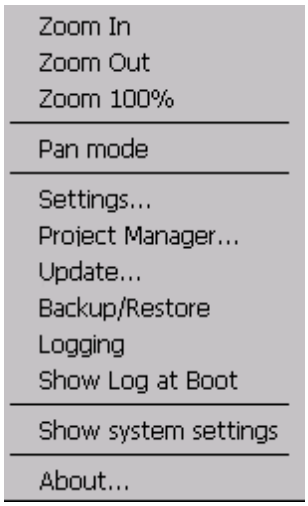


Figure 216

When Backup is called from the context menu, a dialog appears to guide the user in backup operation selecting the path where to save the .zip file with backup.

Figure 217

Backup files can be saved in all available storages like USB, SD card, network folders etc.

NOTE *Backup is available in WCE only platform. It is not supported in Win32 / JMobile Client.*

NOTE *Backup ignores external files stored on USB and SD cards. So, if dynamic data like recipes, trends, events are archived there the backup will ignore it.*

Backup package can be restored from a formatted HMI panel using **Transfer from disk** option in the BSP Loader menu. Just select backup file and the system will automatically check the package to confirm its compatibility with the current platform and install it.

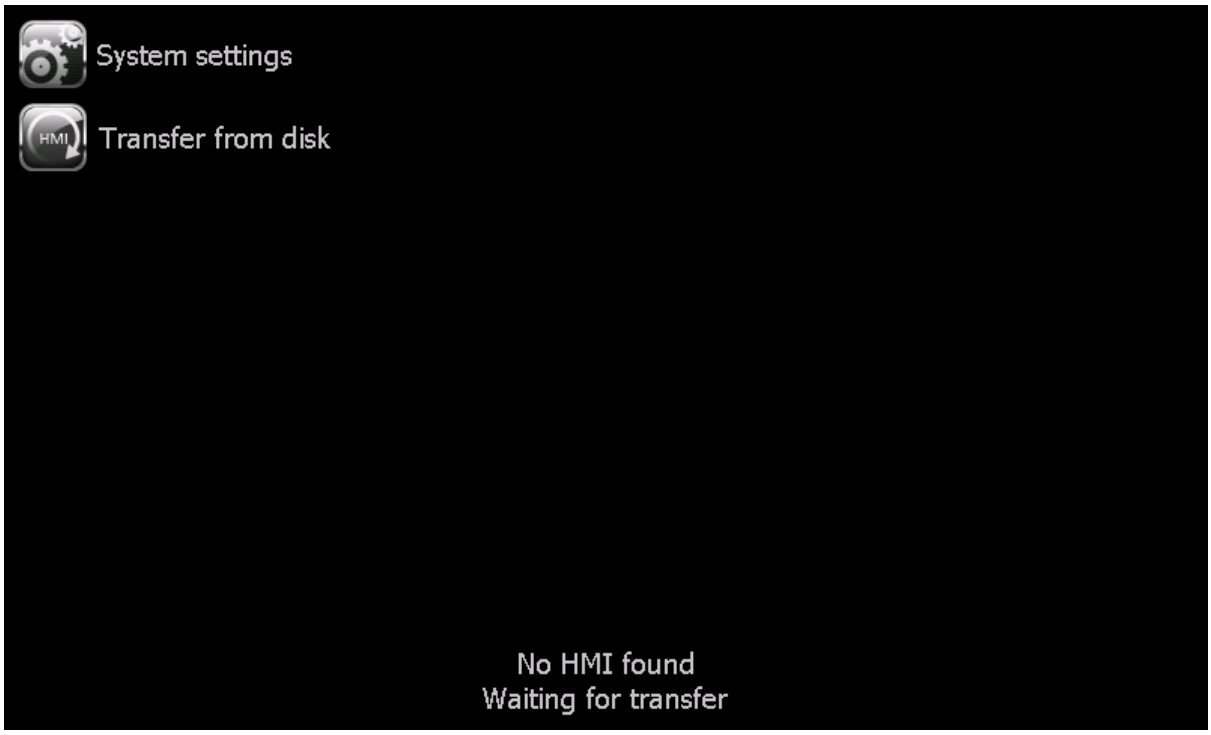


Figure 218

27 Keypads

Keypads are used for data entry operations. Several keypads are provided in the Studio by default, including Numeric, Alphabet, Alphabet Small and Up-Down, as shown in the following pictures:

Figure 219

Figure 220

Figure 221

27.1 Creating and Using Custom Keypads

Keypads can be created from scratch using the following procedure. Note that you can also change the existing keypads.

From the Project View pane right-click on the Keypads folder. A context menu will be displayed, as shown in the figure below:

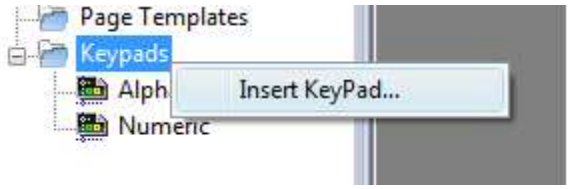


Figure 222

Clicking on the **Insert Keypad** will generate a pop-up with the **New Keypad** dialog, as shown below.

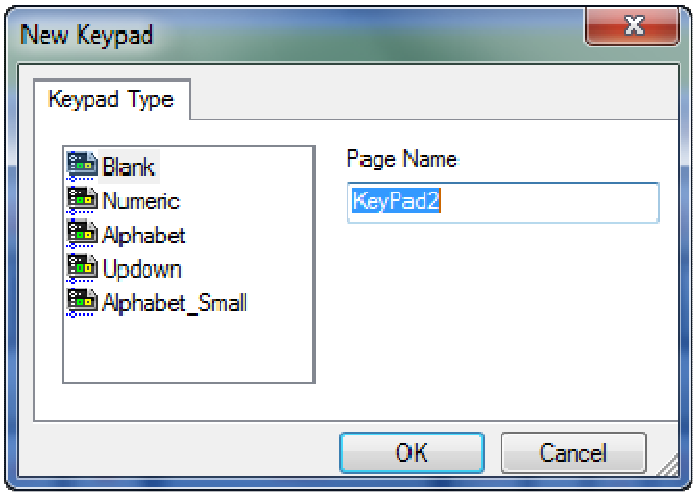


Figure 223

The user can select any of the available keypads that are provided in the project template (the list shown on the left side) to create a custom keypad. If you need to create a keypad from scratch, then select the “Blank" option. This will insert a Blank Keypad, as shown below:

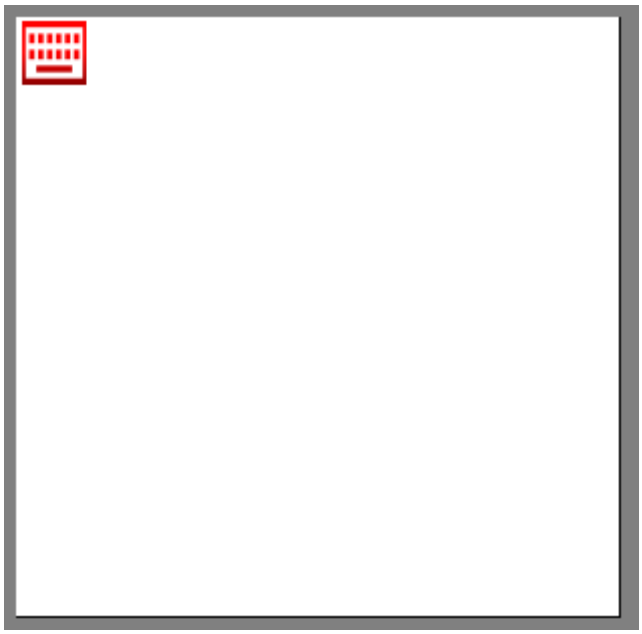


Figure 224.

You can use the widgets available from the **Keypad Widgets gallery** (as shown in the picture below) to create the custom keypad.

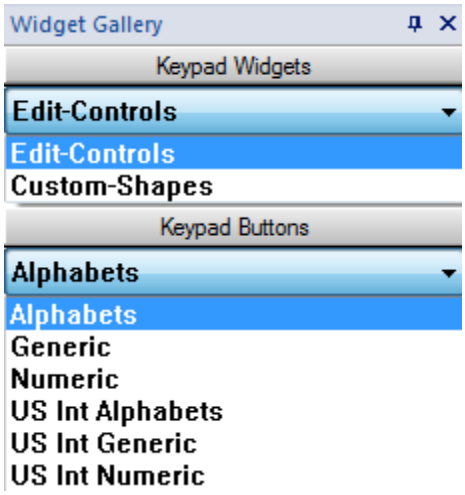


Figure 225

A sample custom-created keypad is shown below. Newly created keypads will be saved in the project folder.

Figure 226

Once the custom keypad has been created, it may be used for any specific field where the Keyboard Type property has been properly set, by selecting the corresponding keypad from the property **Keypad Type** in the property pane as shown below.

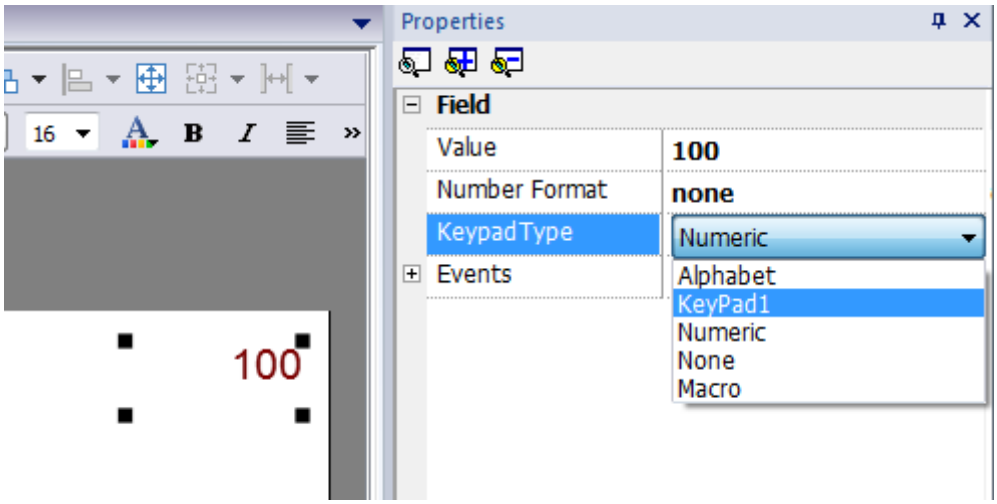


Figure 227

The Up-Down keypad is mainly used for moving cursors in Widgets that are requiring this function. An example is the “Control List" as shown in the following picture.

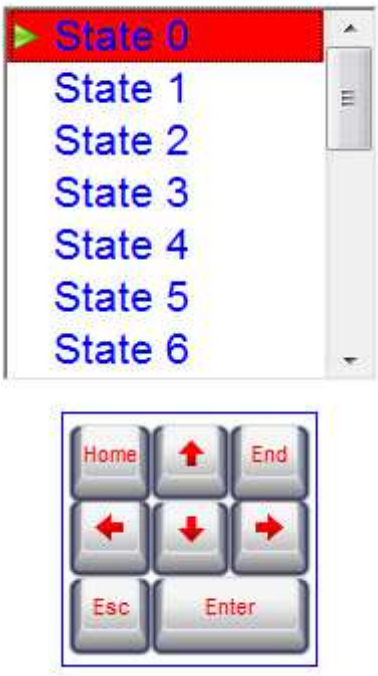


Figure 228

27.2 Deleting or Renaming Custom Keypads

In the Project View pane right-click on the keypad you need to delete or rename. A context menu will be displayed as shown in the figure below.

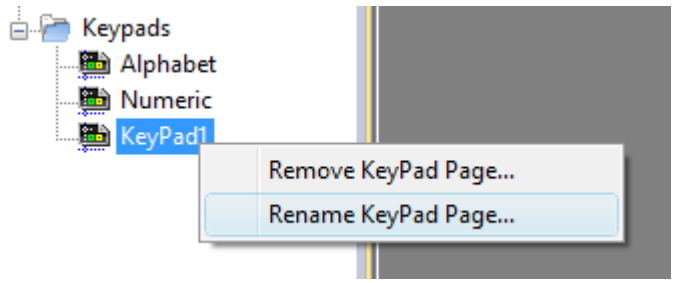


Figure 229

The user can choose the options:

- **Remove KeyPad Page** to remove the keypad from the project
- **Rename Keypad Page** to rename the keypad.

NOTE *By default, any numeric widget (read/write numeric field) will be assigned the numeric keypad. If you decide to modify the default numeric keypad that will be used throughout the project, the following procedure is recommended, so you won't need to assign that new keypad to all numeric entry widgets. First, create a new keypad, using the numeric keypad as the keypad type and save it with a different name. This will be a backup of the numeric keypad. Then open and modify the default numeric keypad, and save it with its original name. The now modified numeric keypad will be assigned by default to all numeric fields in the project.*

27.3 Keypad Type

The **Keypad Type** is one of the parameters available in properties window of keypads. Use this parameter to define what type of data entry is needed. Follow the list of options available:

Auto	This is the default.
Decimal	Only numeric keys are accepted. Entering 10, the keypad return back value 10 that will be display as 10 if the attached field is numeric or ASCII, as A if the attached filed is hexadecimal.
Hexadecimal	Only hexadecimal keys are accepted. Entering 10, the keypad return back value 16 that will be display as 16 if the attached field is numeric or ASCII, as 10 if the attached field is hexadecimal.
Ascii	All keys are enabled. Entering 1A keypad return back value 1A that will be display as 1 if the attached field is numeric, as 1A if the attached field is ASCII or as 1A if the attached field is hexadecimal.

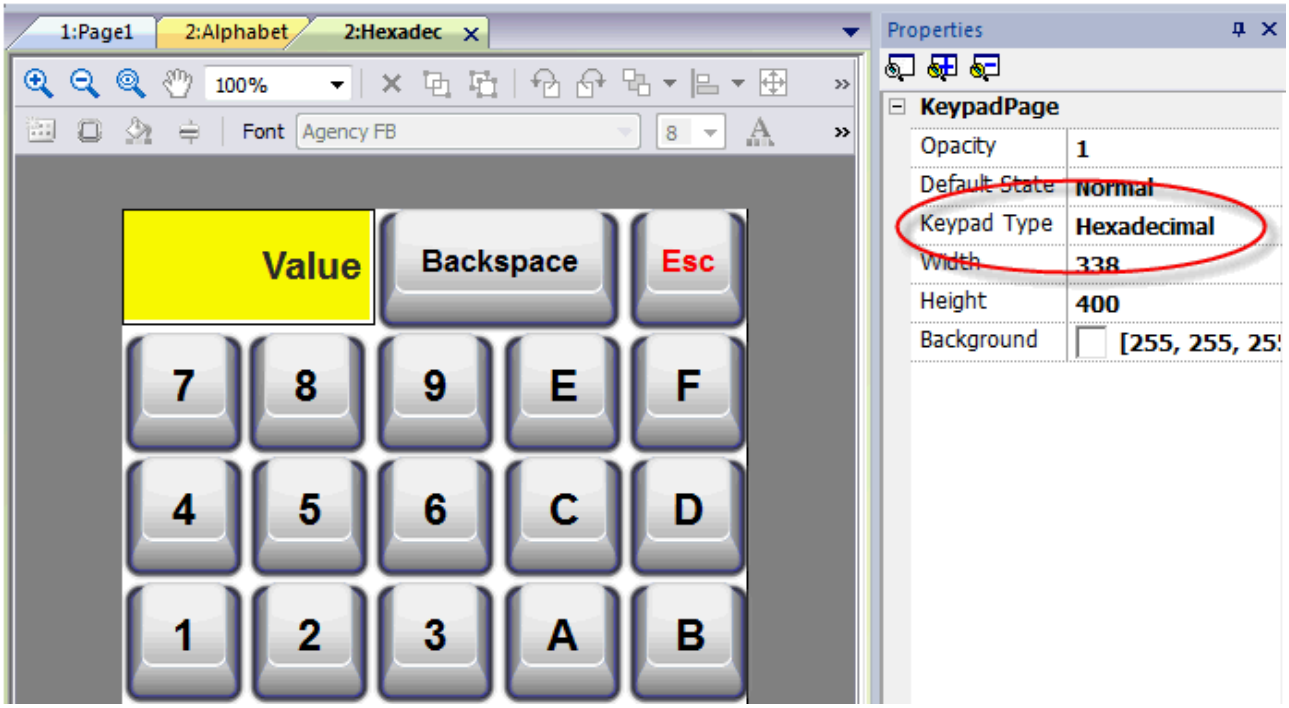


Figure 230

27.4 Keypad Position

Runtime Positioning property of keypads can be used in JMobile Studio to define where keypads will appear in the screen. The following options are available:

- **Automatic**: runtime will show keypads in the best position based on widget position where data entry is required.
- **Absolute**: user can define screen x,y position of keypad (in pixel).
- **Left-top** | **Left-center** | **Left-bottom** | **Center-top** | **Center-center** | **Center-bottom | Right-top** | **Right-center** | **Right-bottom:** predefined screen positions.

As default keypads can be moved using dragging. Use **Lock keypad position** to disable it if required.

28 External keyboards

Runtime has been designed to work with external keyboards connected via USB.

Keyboards can be used for:

- Data Entry (default)
- Actions map on specific keys

You can map for example the “right arrow” key event "OnClick" to the **LoadPage** action.

You can configure your keyboard at project level so that the setting you create will be inherited by all the pages. In each page you can then choose which key setting will be inherited from the project and which one you will customize for the specific page.

The Keyboard Editor can be opened using the tab **Keyboard** at the bottom of the project or page workspace.

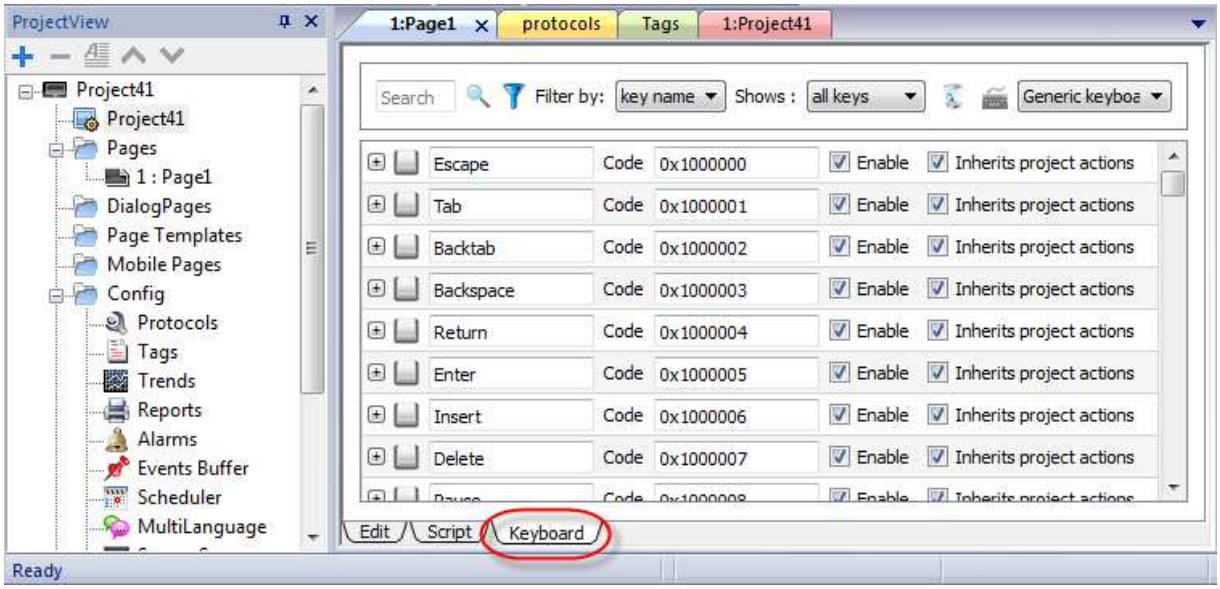


Figure 231

Each row in the Keyboard Editor corresponds to a Key. For each key, the following information is available:

Item	Description
Label	The name of the key
Code	The code of the key
Enable	The individual enable status of the key
Inherits project actions	Defines whether the key is inheriting the action programmed at the project level

The table shows the possible configurations:

Enable	Inherits project actions	Editor appearance	Runtime behavior
Checked	Unchecked	Action lists show the page actions (or nothing if the list is empty)	Only the page actions (if any) will be executed.
Checked	Checked	Action lists show the project actions only and cannot be edited	Only the configured project actions (if any) will be executed.
Unchecked	Checked	**Inherits project actions** checkbox and all action lists are disabled. Action lists show the project actions only.	No page or project action will be executed.
Unchecked	Unchecked	**Inherits project actions** checkbox and all action lists are disabled. Action lists show the project actions only.	No page or project action will be executed.

28.1 Search and Filter

After selecting **Filter by** to **key name**, you can start typing the name of the Key in the box at the left of the toolbar and the Keyboard Editor will list only the keys whose Label contains the text you have entered.

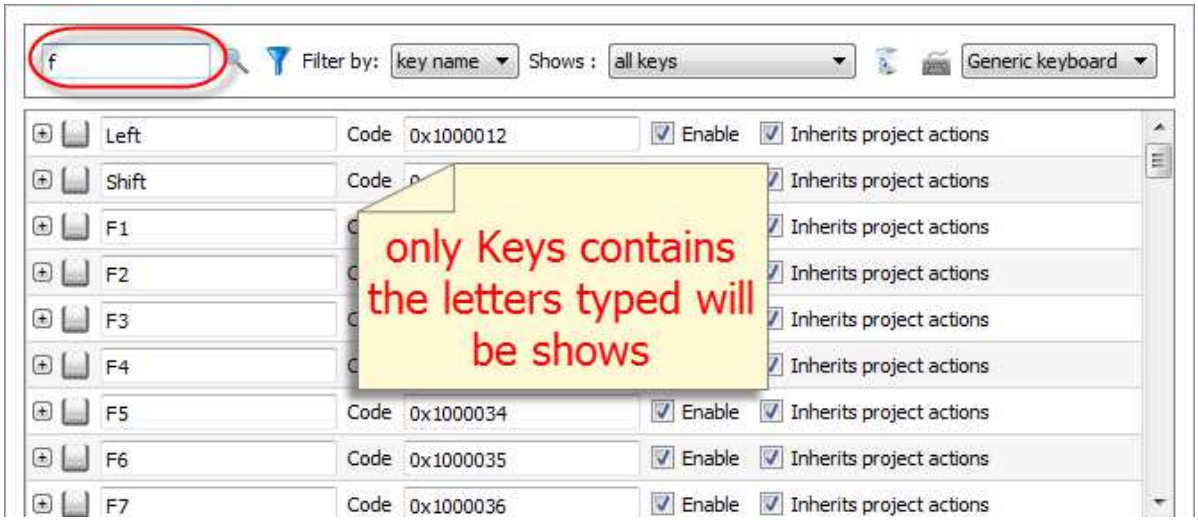
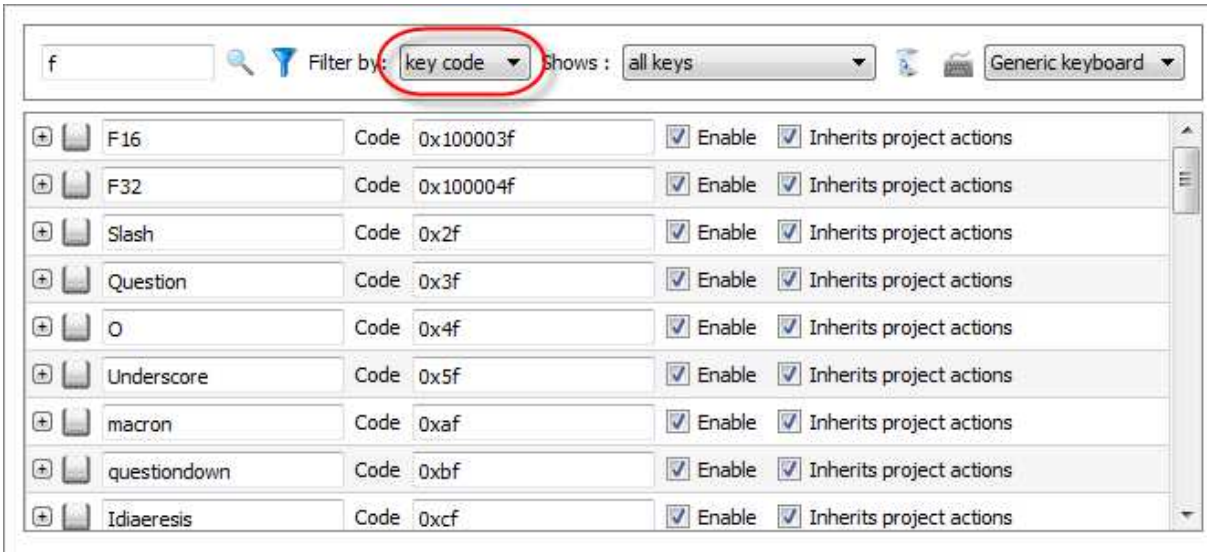


Figure 232

Alternatively, if **Filter by** has been set to **key code** only the Keys contains the text in their **Code** column will appear in the list.


f
Filter by:
key code
Shows :
all keys
Generic keyboard
F16 Code 0x100003f Enable Inherits project actions
F32 Code 0x100004f Enable Inherits project actions
Slash Code 0x2f Enable Inherits project actions
Question Code 0x3f Enable Inherits project actions
O Code 0x4f Enable Inherits project actions
Underscore Code 0x5f Enable Inherits project actions
macron Code 0xaf Enable Inherits project actions
questiondown Code 0xbf Enable Inherits project actions
Idiaeresis Code 0xcf Enable Inherits project actions
Figure 233

28.2 Shows

You can easily select what keys will be listed in the Keyboard Editor window.

Code	Description
All Keys	The editor will show all keys available in the keyboard layout
Modified Keys	The editor will show only keys that have been configured with some actions at the page level
Modified Keys in project	The editor will show only keys that have been configured with actions at the project level

28.3 Clear Actions

The **Clear Actions** button is available to delete actions configured for one or more keys in the Keyboard Editor. To Clear Actions, select one or more keys and then press the **Clear Actions** button.
You will clear all actions configured for the selected keys either in the page or in the project. Actions will be removed depending on what you are currently configuring, either the project actions in the project view or the page actions in the page view. A confirmation dialog will appear to request confirmation of the requested command.

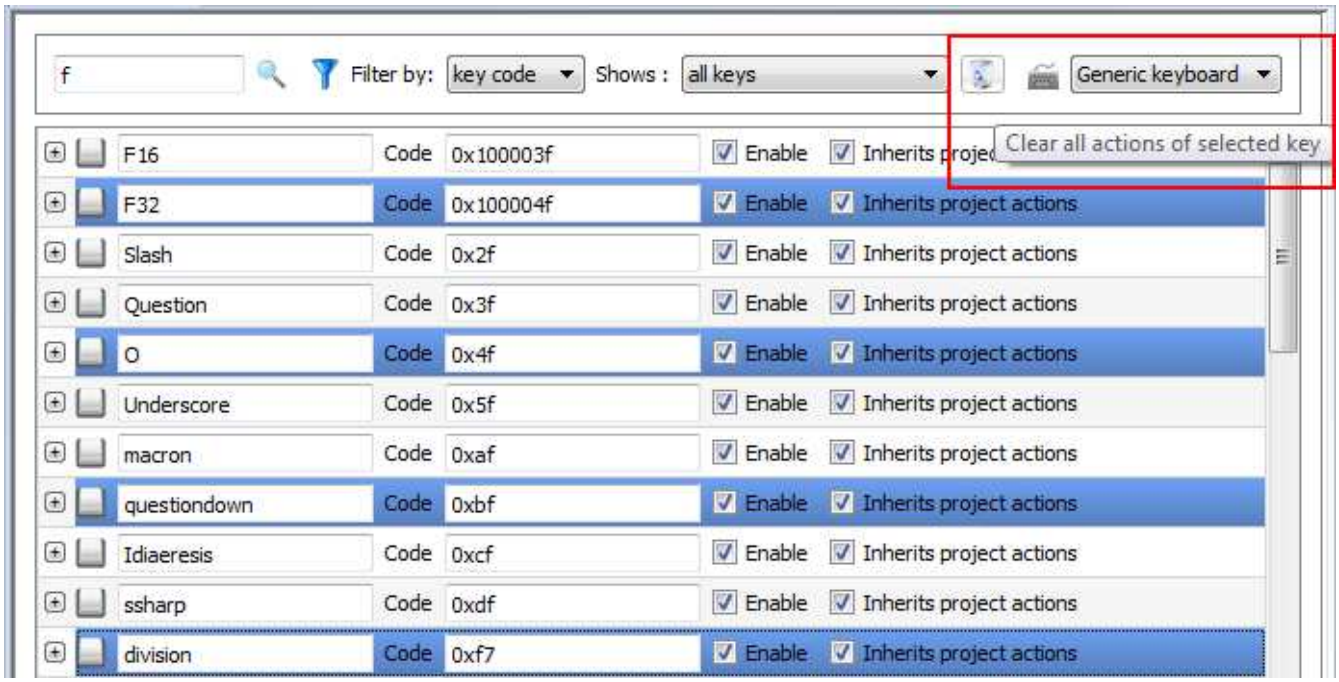

f
Filter by:
key code
Shows :
all keys
Generic keyboard
Clear all actions of selected key
F16 Code 0x100003f Enable Inherits proje
F32 Code 0x100004f Enable Inherits project actions
Slash Code 0x2f Enable Inherits project actions
Question Code 0x3f Enable Inherits project actions
O Code 0x4f Enable Inherits project actions
Underscore Code 0x5f Enable Inherits project actions
macron Code 0xaf Enable Inherits project actions
questiondown Code 0xbf Enable Inherits project actions
Idiaeresis Code 0xcf Enable Inherits project actions
ssharp Code 0xdf Enable Inherits project actions
division Code 0xf7 Enable Inherits project actions

Figure 234

28.4 Keyboard Layout

The Keyboard Layout combo box allows the user to select the layout of the keyboard.
Generic Keyboard corresponds to a generic International Keyboard layout.

28.5 Enable Keyboard

You can enable/disable keyboard actions both at the project level and at the page level. A dedicated property is available in the project property sheet and in the page property sheet.

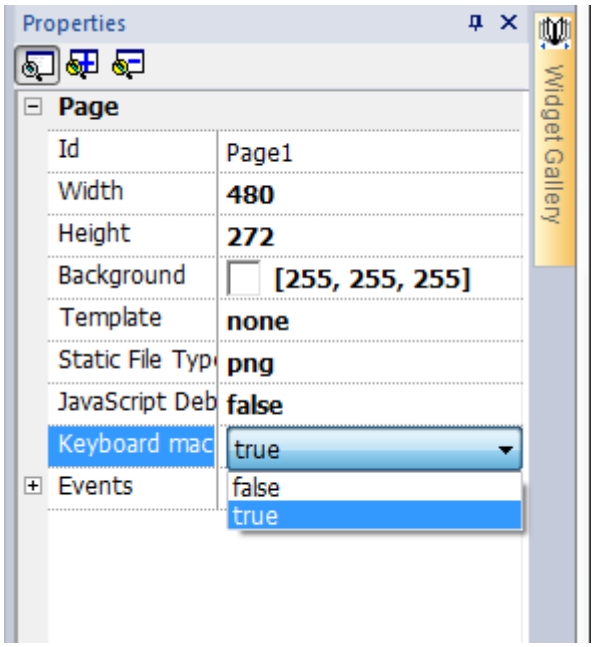


Figure 235

The Keyboard can also be disabled at runtime by using a dedicated macro command **KeyboardMacros**.

28.6 Configure Macro Actions for Keys

To configure actions for keys in the Keyboard Editor just click on **+** on the key you want to program and you will obtains the expanded view for key configuration.

Press the + or JS buttons to add macro commands or Javascript functions to the key event you want to configure.

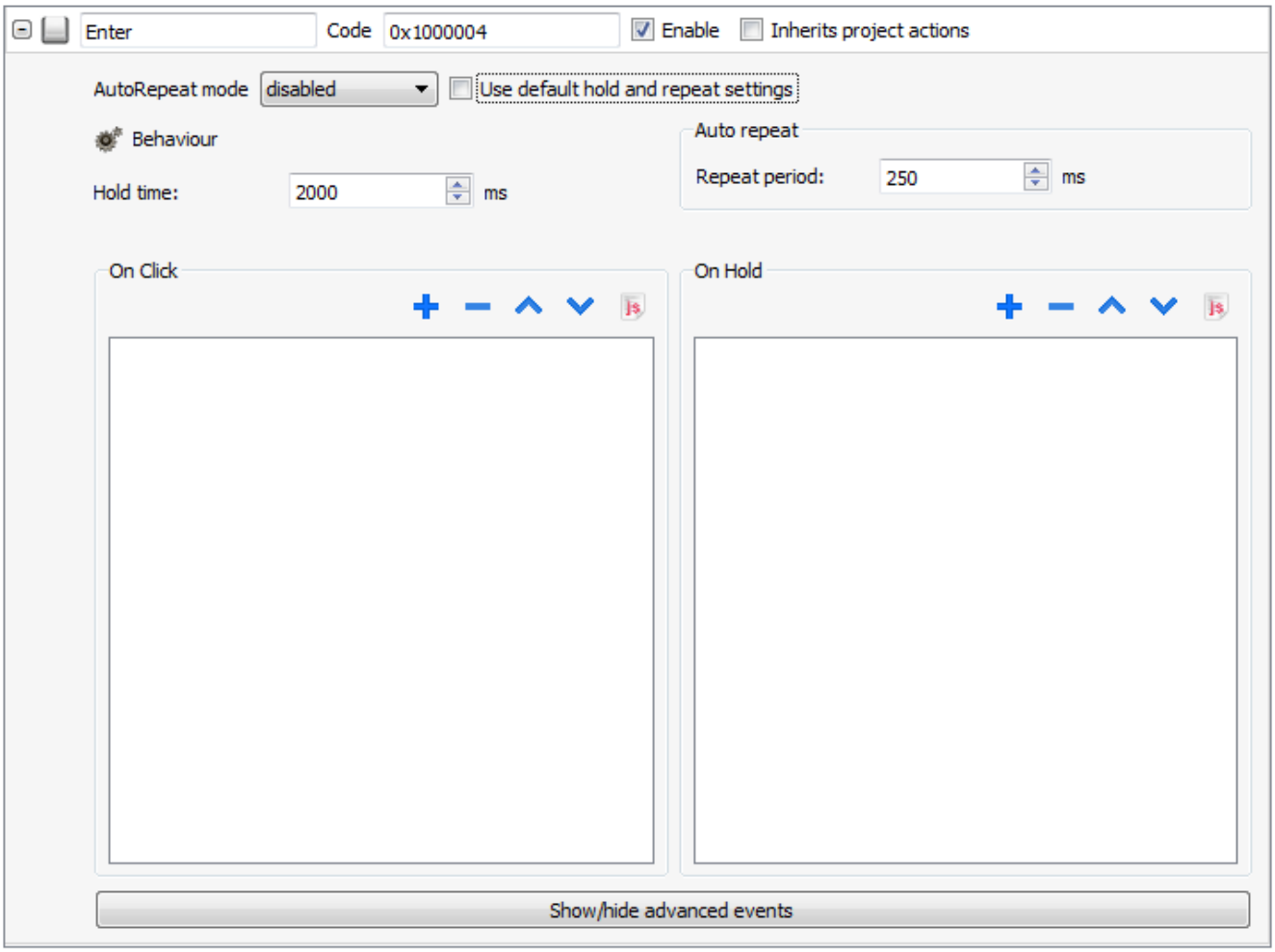


Figure 236

29 Tag Cross Reference

The Tag Cross Reference displays listsa list of Tag Names used in currenta project. based on their location and use..
Using the Tag Cross Reference in JMobile Studio it is possible to:

- Verify where each tag is used (alarms, pages, recipes, schedulers, trends etc)
- Identify invalid tag references (references to tags not existing / not defined in the tag editor)
- Identify tags not used in the project

NOTE *The Tag Cross Reference does not search for tags and related references used in Javascript code.*

29.1 Accessing Tag Cross Reference

The Tag Cross Reference can be accessed by selecting Tag Cross Reference from the left side toolbar of JMobile Studio. Once selected, a **Tag Cross Reference** window will open as shown below.

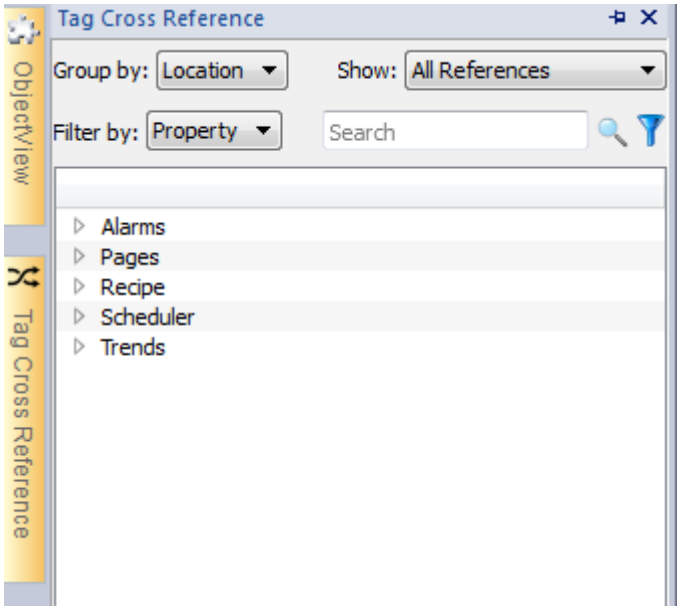


Figure 237

29.2 Using Tag Cross Reference

The Tag Cross Reference window provides the following:

- **Group by:** allows to choose if group tags informationsare grouped by **Location** (alarms, pages, trends etc) or by **Tag** name
- **Show:** selects content to shown. Options available are **All Reference** (all tags), **Invalid Tag Reference** (usually tags missing in tag editor) and **Unused Tags** (list of tags defined in Tag Editor but not used in the project).
- **Filter by/Search filter:** allows applying a search filter to the selected content. Filter can be applied to **Location** (ex. of locations are “Alarms”, “Pages”, “Recipe”), **Tag** (tag name) or **Property** (ex. “value”, “fill”, “visibility”).

All data is results are organized in a tree that users can browse to search where the tags are used in the project. Double click on resources inthe item inside the tree to open/select the resource in the project.

29.3 Tag Cross Reference: data update

The Tag Cross Reference collects and organizes information in the background when the project is opened.

Usually dataBy default, the information inside the Tag Cross Reference pane areis not updated every time something changed in the project. Using the reload button (available in Tag Cross Reference bottom toolbar) the user can force an update/refresh of the data. The same reload button appears with a “!” when a refresh is needed.

Figure 238

Users can choose to enable Auto Update by checking **Tag Cross Reference** -> **Auto Update** from the top toolbar of JMobile Studio into *View -> Properties* dialog.

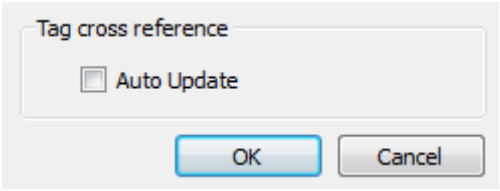


Figure 239

29.4 Export data in csv

Tag Cross Reference allows exporting all information collected in a csv format. The format of the csv file depends on *Group by*the current selection in the Tag Cross Reference and can be one of the following:

- by **Location**: RESOURCE, RESOURCE DESC, WIDGET-ID, ATTRIBUTE, TAG
- by **Tag**: TAG, RESOURCE, RESOURCE DESC, WIDGET-ID, ATTRIBUTE

NOTE *Separators used in export operation depends on regional settings of PC.*

30 Indexed Addressing

Indexed addressing provides a simple way to select a set of tags to use depending on the value of another tag.

Indexed addressing comes very handy in re-using the same graphics to visualize a set of data coming from different homogenous sources, just by letting the user pick the machine to monitor from a list.

Graphic re-usage is also very important in achieving better overall performances.

30.1 Creating an Indexed Addressing Set

The following example introduces the common usage scenario of Indexed Addressing.

The sample system consists of a building with 4 rooms, and each room is equipped with Temperature, Pressure, and Humidity sensors. All data can be laid out in a table form:

Room Number	Temperature	Pressure	Humidity
1	Room1-Temperature	Room1-Pressure	Room1-Humidity
2	Room2-Temperature	Room2-Pressure	Room2-Humidity
3	Room3-Temperature	Room3-Pressure	Room3-Humidity
4	Room4-Temperature	Room4-Pressure	Room4-Humidity

Taking advantage of the Indexed Addressing feature, you can use the same table form to easily arrange your data in the HMI.

All the three different sensors can now be visualized in a single page like the following, in which Room number is used as a selector (combo box) to pick up the right set of tags.

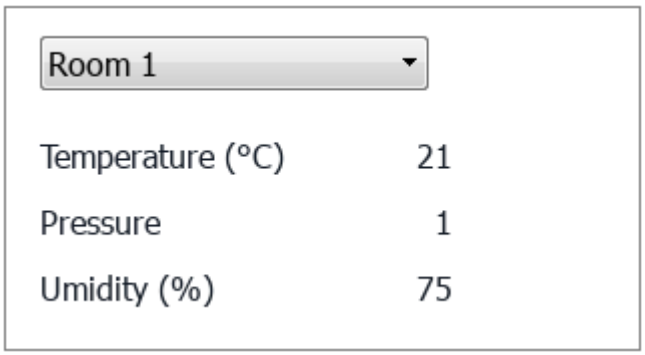


Figure 240

In order to create the **Indexed Tag Set** of the previous example in JMobile Studio.

1) Define protocols and tags in Tag Editor. For each room and each sensor to visualize an appropriate tag is required.

Name	Group	Driver	Address
Room1-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400001 unsignedShort
Room1-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400002 unsignedShort
Room1-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400003 unsignedShort
Room2-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400004 unsignedShort
Room2-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400005 unsignedShort
Room2-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400006 unsignedShort
Room3-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400007 unsignedShort
Room3-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400008 unsignedShort
Room3-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400009 unsignedShort
Room4-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400010 unsignedShort
Room4-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400011 unsignedShort
Room4-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400012 unsignedShort

Figure 241

2) Create a tag *RoomNumber* to use as **Index Tag**, for selecting rooms. In this example could be of type *UnsignedInt* using *Variable* protocol.
3) From *ProjectView*, double click on *Config* -> *Tags* -> **Indexed Tag Set** to open Indexed addressing configuration page.
4) Click on "**+ Add**" button to create a new Indexed Tag Set.
5) Assign Indexed Tag Set name "*Room"*
6) Select tag to use as selector for rooms "*RoomNumber*"
7) Create 4 **Index Instance** (rooms), one for each room.
8) Create 3 Alias (variables) as table columns and rename it to "*Temperature, Pressure & Humidity*" using double click on table columns
9) Double click on cells of table to select tags as shown below (or click **F2** on cells for entering in edit mode)

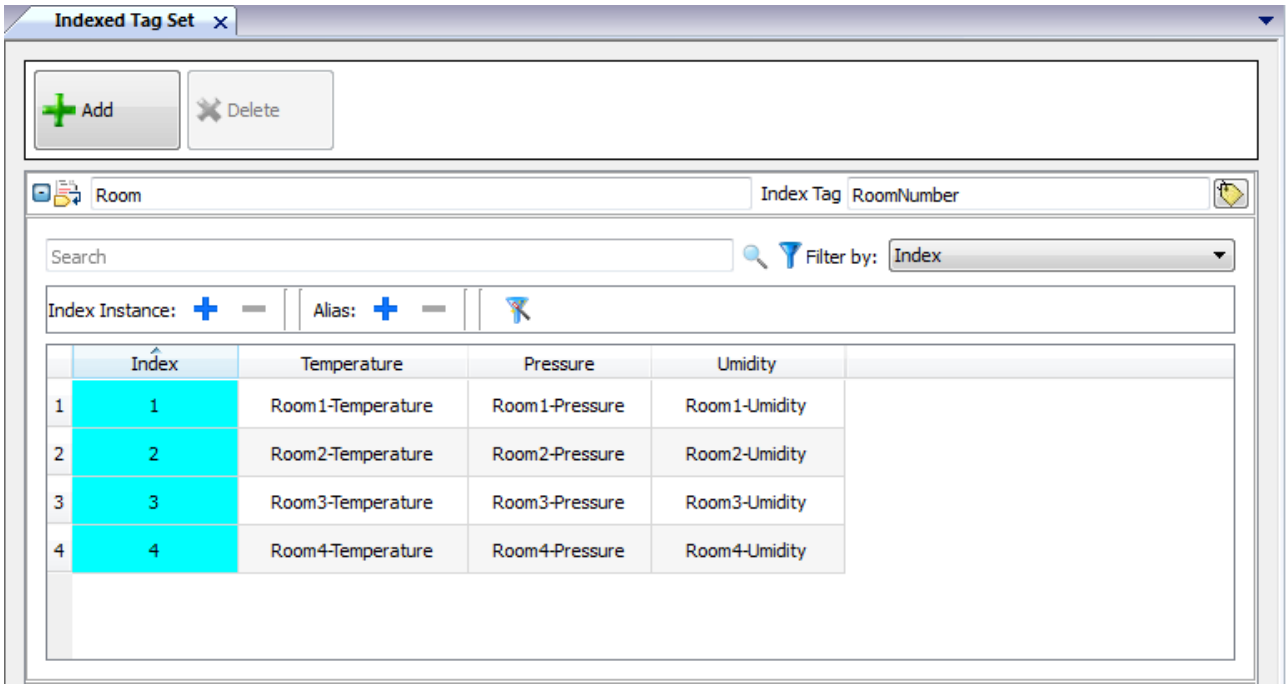


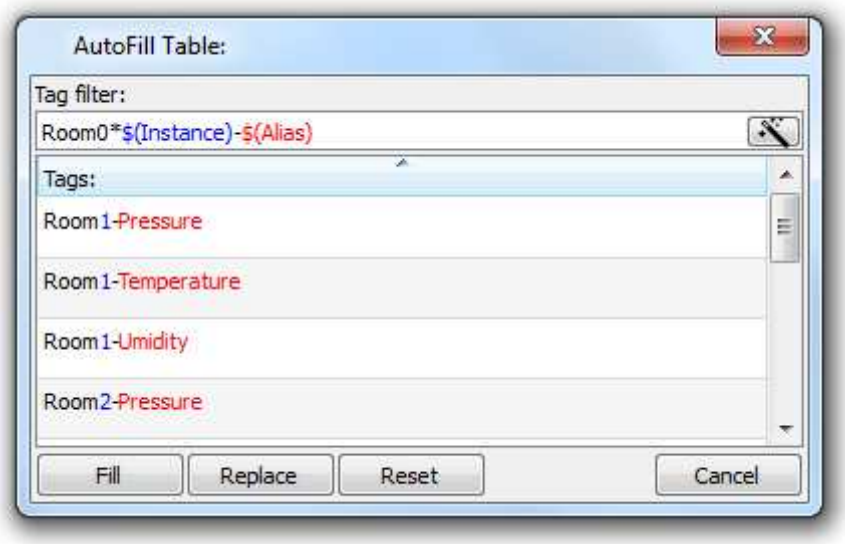
Figure 242

NOTE ***Index Tag*** *datatype can be a number, a string or any type of simple data types.*

30.1.1 Autofill tag names

The process of manually filling all required table entries can require a lot of effort and can be affected by typing errors.

To speed it up, an autofill feature is available in the toolbar .
AutoFill only requires setting *Index* instances and *Alias* names, that are consistent with tag names. In our example, the index column must be filled with room numbers 1...4 and columns names Temperature, Pressure and Humidity.



AutoFill uses regular expression for populating the table with tags trying to match $(Instance) (room number) and $(Alias) (sensor) inside tag names. For example Room$(Instance)-$(Alias) will match all tag names: Room1-Temperature, Room1-Pressure, Room1-Humidity, Room2-Temperature, etc.

Regular expressions can be edited manually when needed, and, more conveniently, you can use the wizard button to let the system guess a good filter expression for your application.

Fill button will fill in missing entries in the tag table using the set filter (if any). For example when new instances or new aliases are added you can use this option to just fill in the new missing entries.
Replace button can be used to replace all table entries with the ones provided by the AutoFill table.
Reset button reset *Tag filter* to empty (no auto fill).

Filters are saved as project preferences and can be separately set for the entire table or for a column. Once set for a column, the table filter is discarded in favour of the column one. So you can selectively change the filter for handling a particular alias only.

30.2 Using Indexed Addressing mode in pages

Once defined an Indexed Tag Set, let see how to design the HMI page introduced in the previous example.

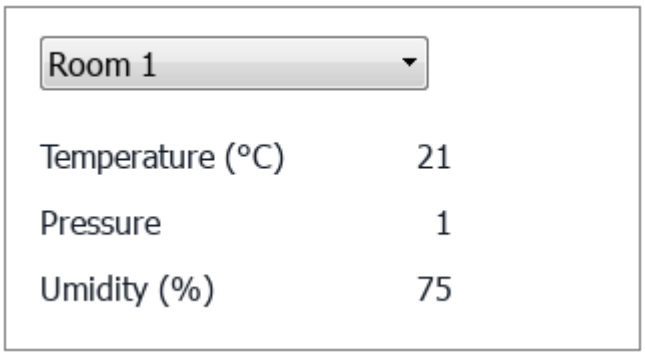


Figure 243

To create this Indexed Tag Set page proceed as follow:

1) Create a page and add a Combo box widget, 3 labels and 3 numeric fields.
2) As *Index* of Combo box will be used the tag *RoomNumber* (**Index Tag**), so the selector of the room.
3) As *List* of Combo box will be used following.

Index	String List
0	Room Number
1	Room 1
2	Room 2
3	Room 3
4	Room 4

4) Attach to each numeric field *Value* the corresponding **Alias** variable (*Room -> Temperature*, *Room -> Humidity*, *Room -> Pressure*).

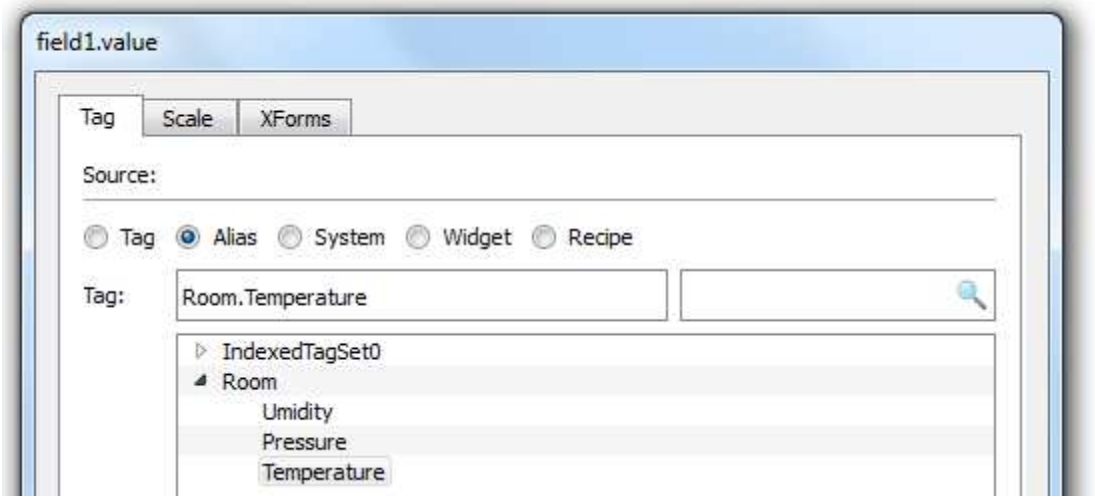


Figure 244

31 Special Widgets

31.1 Date Time Widget

The Date Time widget is a widget that can be used to view and edit the current time and date at Runtime. The widget can be found in the Widget Gallery.

In the Property pane of the widget you can set the format of the date and time as “Date only”, “Time only”, “Date and Time”. Different formats for representing date and time are available as shown in the figure below.

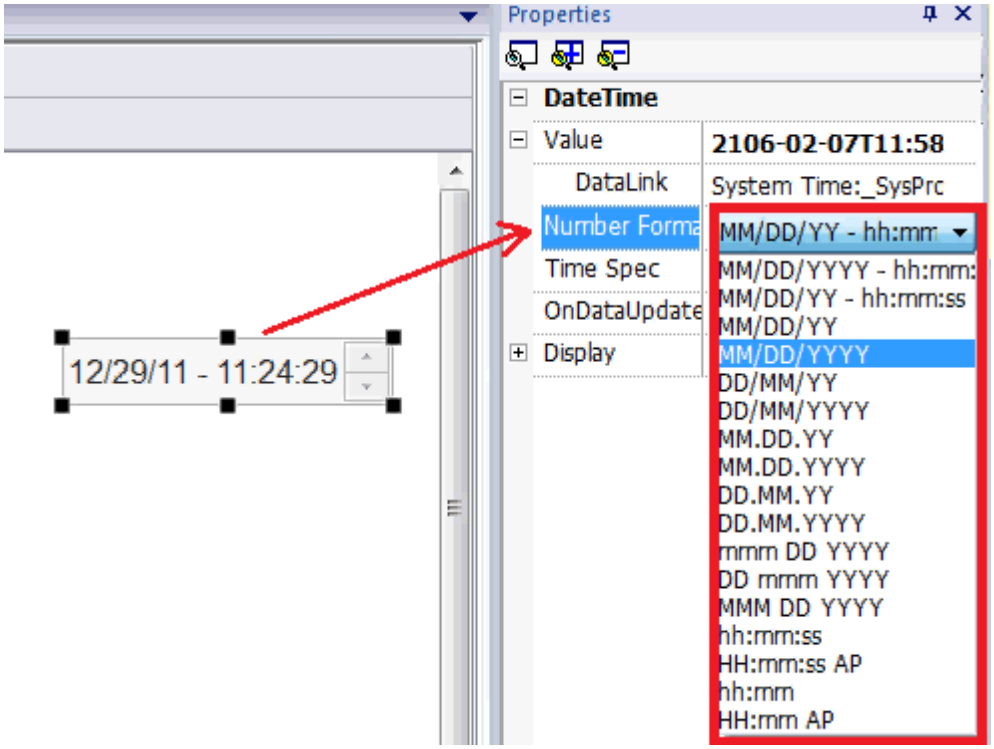


Figure 257

The Time Spec option allows selecting which time the widget has to show during Runtime; three options are available for this property:

- Server
- Local
- Global

To understand the difference between the options available for the “Time Spec” property, you need to recall the basic concepts behind the HMI system architecture. Please read the chapter “Runtime Modes” to become familiar with the HMI software architecture first.

If you select “Server” as Time Spec, the widget will show the time information as handled by the server side of the HMI system.
If you select Global as Time spec, it will show the Global Time (GMT).
If you select Local as Time Spec, it will show the Local Time in the Widget (the time of the target where the project is running).

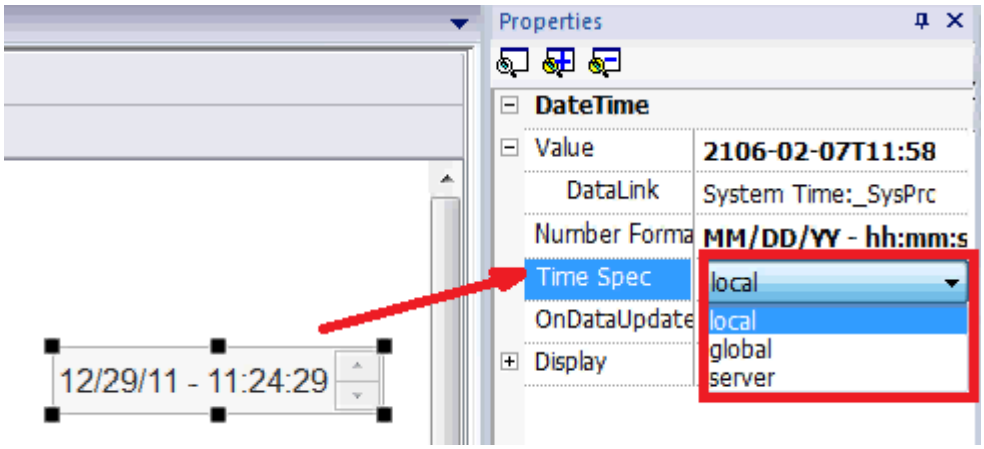


Figure 258

31.2 RSS Feed Widget

The RSS (**R**eally **S**imple **S**yndication) Feed widget allows you to display on the screen your favorite RSS feeds directly from the Internet.
The widget is available in the Widget Gallery. When placed on the page the widget looks as shown in the figure below

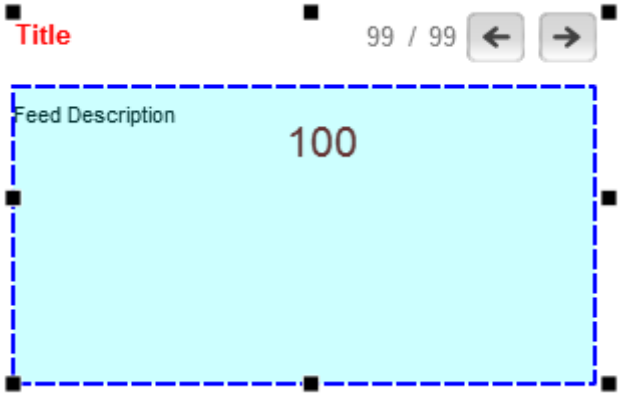


Figure 259

The RSS Feed widget main properties are:

RSS Source
Allows you to specify the feed URL

UpdateRate
Allows you to specify the refresh time.

Properties are shown in the figure below.

NOTE *Feeds sources are fixed and cannot be changed at runtime.*

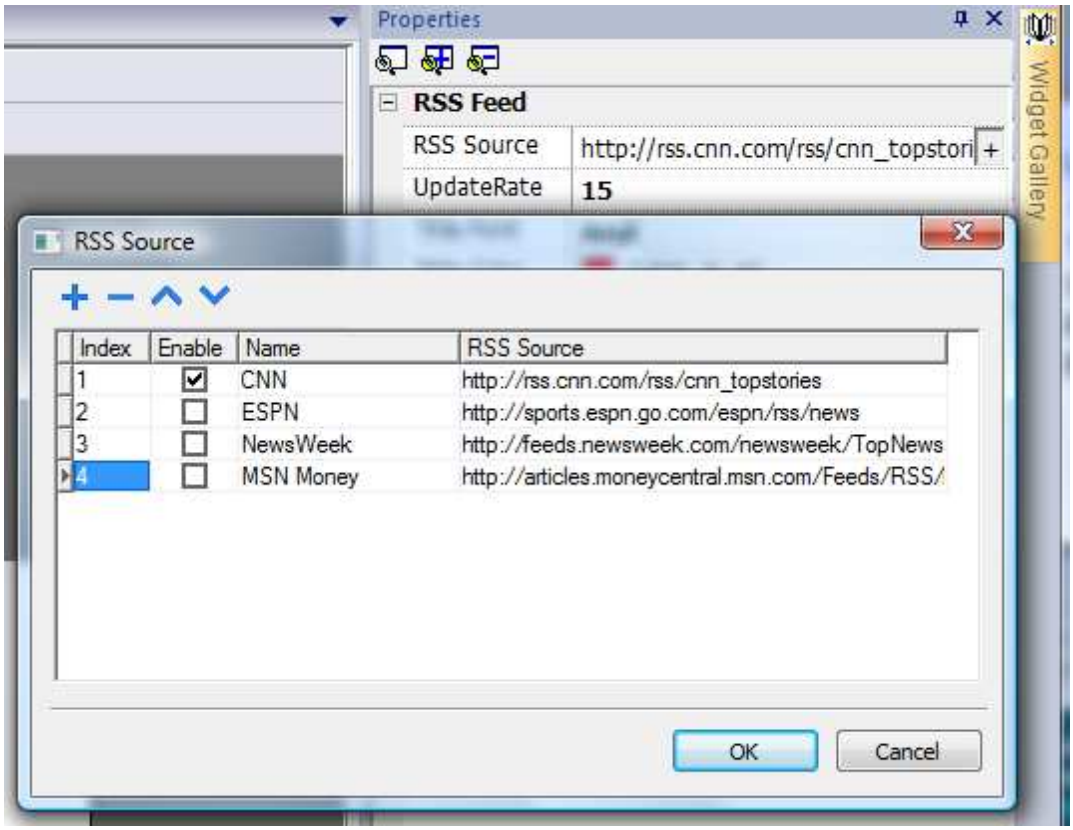


Figure 260

NOTE *The RSS Feed widget is specifically designed to work on units where the Internet Explorer browser (Pocket Internet Explorer) is part of the operating system.*

31.3 Control List Widget

Control List widget is a convenient way to represent the status associated with a particular process, but also a way to control that process from the same widget.
The Control List widget is available in the Widget Gallery.

There are two types of control lists. One is a control list group, in which the up and down buttons are present on the control list itself. The state can be selected with the up and down buttons. The other type of control list has no pre-configured buttons in the group. In that case, the state can be selected by pressing on the screen.

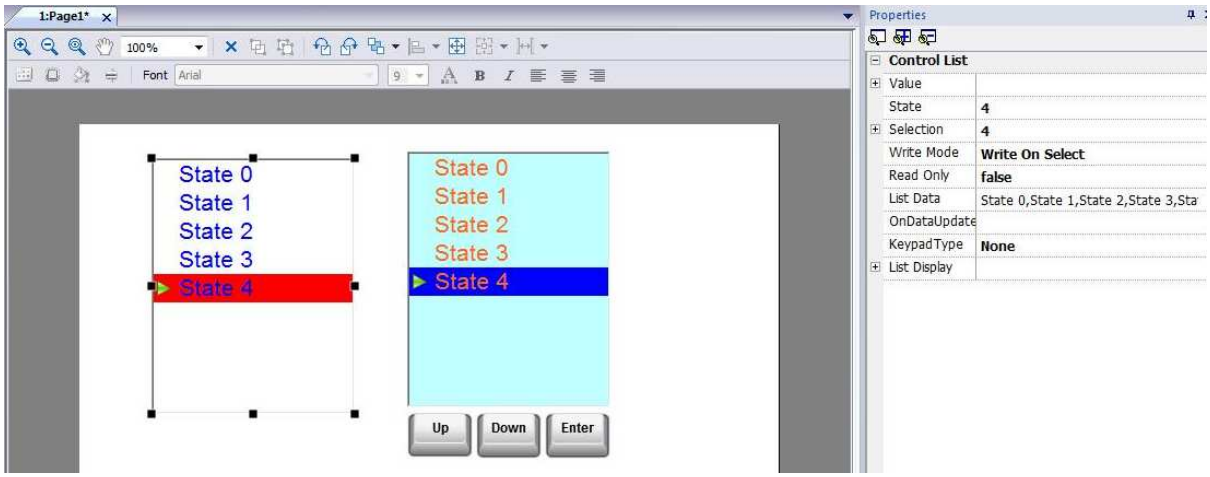


Figure 261

31.3.1 State

States are added by selecting Add/Remove List Items from the List Data option in the property pane. Any value can be assigned to a State; activating the State will result in a write operation to the Tag, which has been linked to the Value property of the Control List Widget.

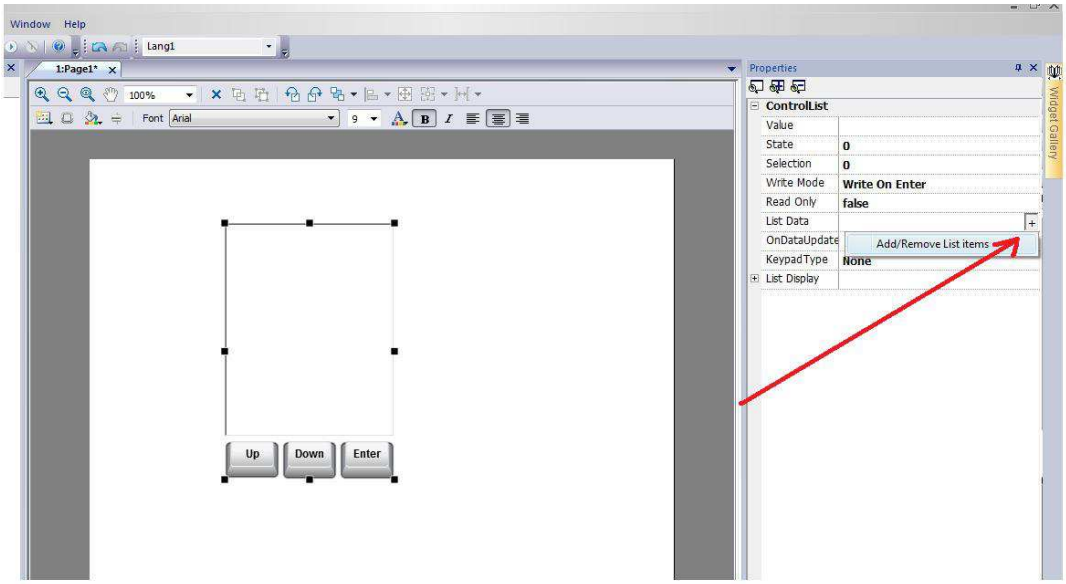


Figure 262

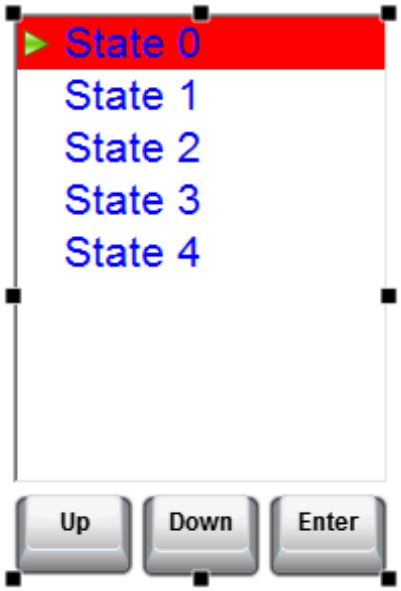


Figure 263

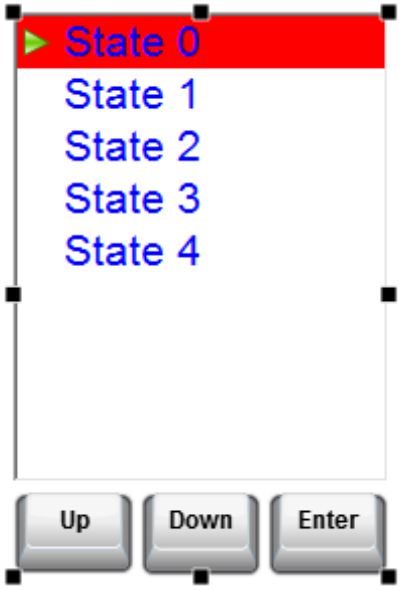


Figure 264

31.3.2 Selection

Selection shows which status is currently selected, and will appear as a highlight cursor moving up and down (according to the use of the defined keys). The Selection property can be attached, as well, to a Tag.
The small triangle on the left side of the list tells you what the current status is.

There are two write modes for the control list: Write on Select and Write on Enter.

31.3.3 Write on Select

On Write on Select, the value will automatically be written when one of the states is selected

31.3.4 Write on Enter

On Write on Enter you need first to select the state, and then press the enter key to write the status value to the Tag.

31.3.5 Read Only

The Read only property of the widget can be attached to a tag and will control whether the control list will be just an indicator, or a combination of both. For example, with a machine in Manual mode, the Control list will let the operator select which state should be active and while in Auto mode, the list is an indication of the active step.

31.4 Variables Widget

The Variables widget is available in the Advanced category under the Data Sources sub-category as shown in the following figure.

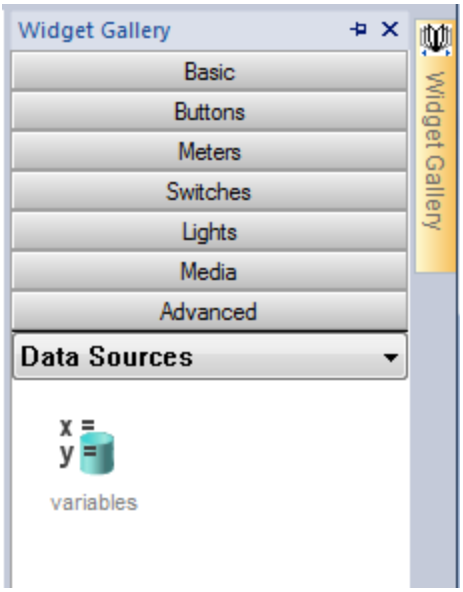


Figure 265

The purpose of the Variables widget is to have some internal variables that can be used for operations such as data transfer or use in JavaScript programs. The variables are local to the page where the widget has been inserted.
To insert the widget in a page, just drag and drop it to any position on the page. This will display a place holder to indicate that the widget is present, but it will not be visible at runtime. You can create some variables and assign values as shown in the following figure.

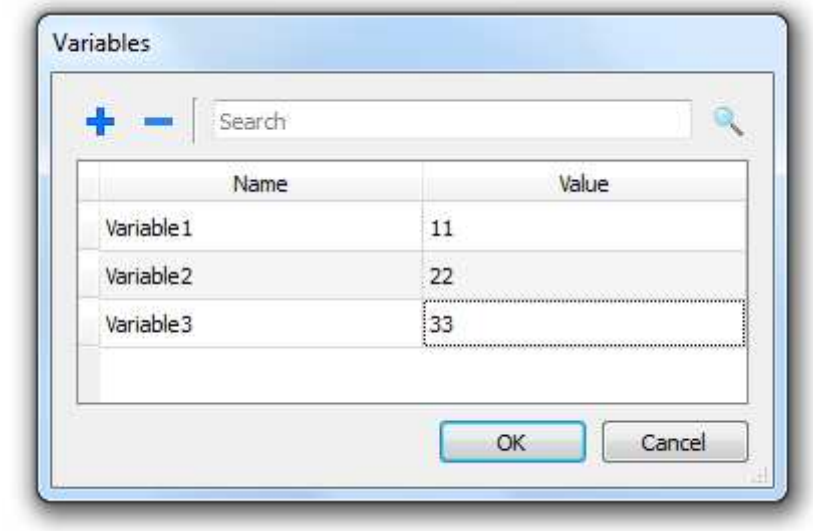


Figure 266

The configured Variables can be referenced from the Attach tag dialog once you click on the Widget source as shown in the following figure.

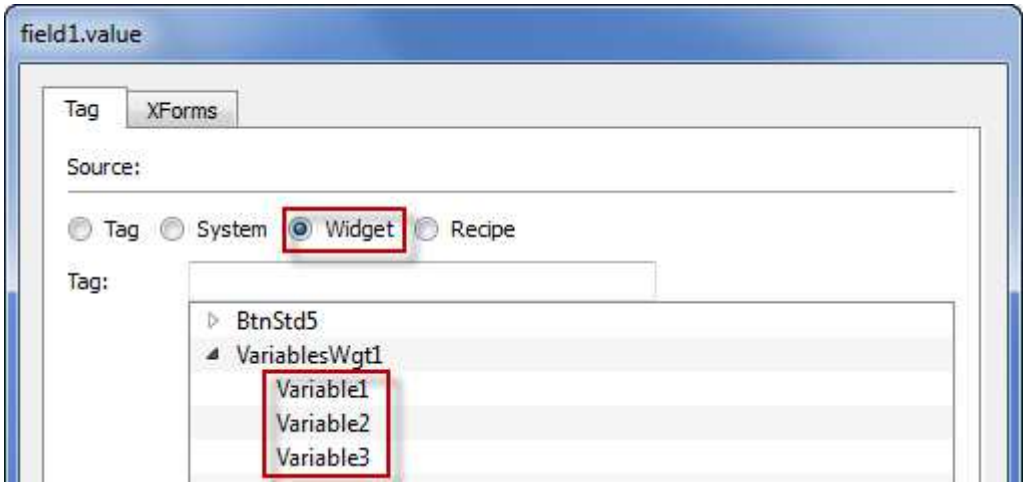


Figure 267

In case global variables are needed, they can be configured from the project widget, adding the desired variables to the global Variable Widget as shown in the following figure

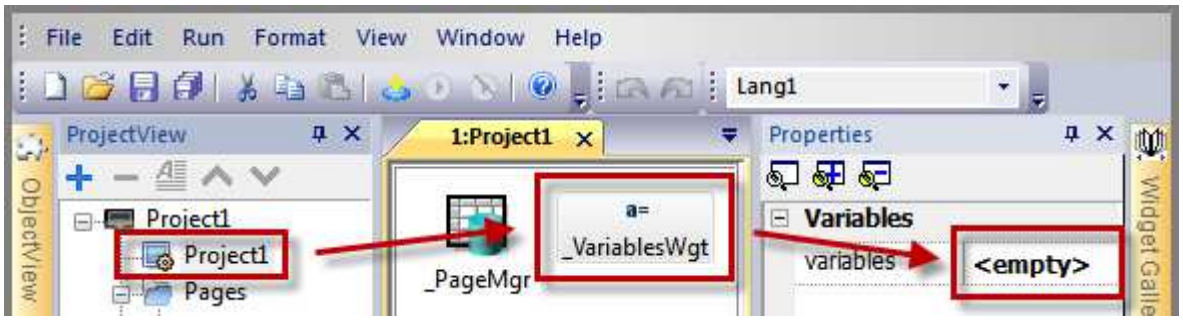


Figure 268

31.4.1 Using Variables in JavaScript

The Variables can be also referenced in JavaScript programs with the following syntax:

For Local Variables

```
var varWgt = page.getWidget("_VariablesWgt");
var compVar = varWgt.getProperty("VariableName");
```

For Global Variables

```
var varWgt = project.getWidget("_VariablesWgt");
var compVar = varWgt.getProperty("VariableName");
```

31.5 Media Player Widget

The Media Player Widgets are available into the widget gallery. The widget can be used to play videos from a playlist. The video files can be stored on the Flash, a USB memory stick or an SD Card. There are two available widgets. The first one is without the multimedia frame so it just plays the video without any user control. The second widget includes a multimedia frame where you can find buttons to play and stop the video.

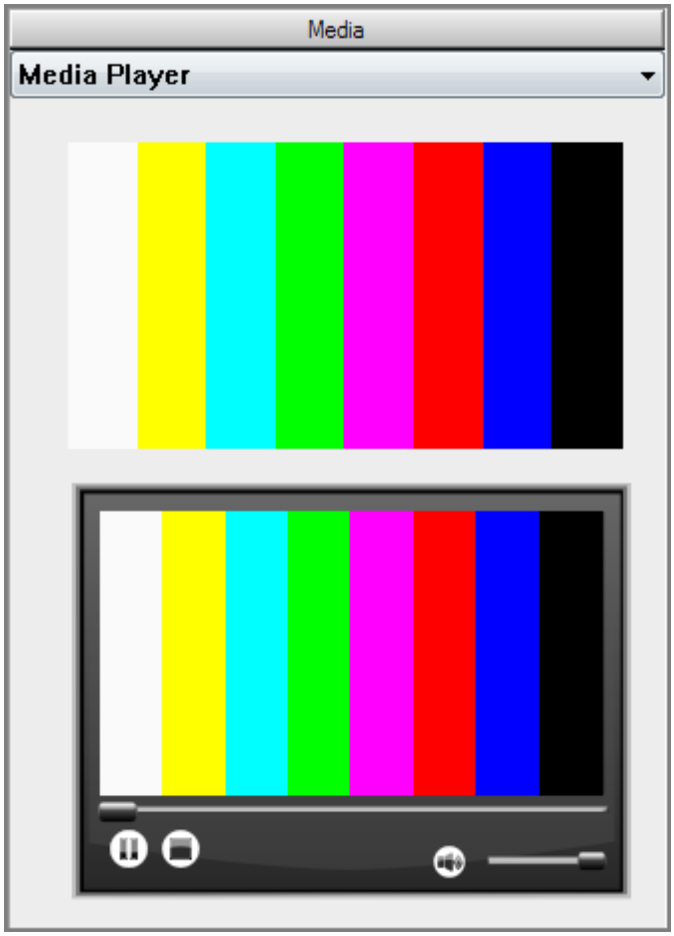


Figure 269

To use the widget just drag and drop it into a project page. The Media Player widget properties are:

Media Player List Allows you to create a playlist by selecting the video files to be available to the widget

Loop Style Allows you to define the way the video in the playlist is played. The available values are **NoLoop**, **LoopOne**, **LoopAll** and **Random**.

- **NoLoop** allows you to play all the videos in the playlist, then it stops.
- **LoopOne** allows you to loop on the first video in the playlist.
- **LoopAll** allows you to loop on the playlist following the list of videos.

Random allows you to play the videos in a random order.

NOTE *The Media Player Widget works only with some HMI panels* (*hmi panels based on ARM Cortex-A8-1Ghz and Win32 platform*). *It doesn't work the* JMobile Client *or ActiveX. You can have only one Media Player widget in a page.*

NOTE *Runtime support up to one video player widget for page. It's recommended also to use same codecs and settings for video of a playlist and not mix it.*

It supports videos encoded with the following codecs:

- DSP based video codecs
  - H264 using AVI/MP4 container, CABAC off and Level 3 (suggested)
  - MPEG2 using AVI container
  - MPEG4 using AVI container

- Software video codec
  - Microsoft MPEG4 v3 using an AVI container.

DSP Video Codecs

The first three codecs use the processor DPS (video hardware acceleration) and you need a BSP 1.55 to play them .These videos can have a maximum resolution of 720*576 pixels and a bit rate of 4200 kb/s. 720p and 1080p are not supported. Audio is not supported.

Software Video Codec

The videos encoded with Microsoft MPEG4 v3 are not using the hardware acceleration and have more limitations. To prevent the videos from running jerky a maximum resolution of 640x512 pixels and a bit rate of 1300kb/s are suggested. In addition, the size of the Media Player widget used on the page should have the same size as the videos in the playlist, in order to avoid upscaling and downscaling. Audio is not supported.

How to convert a video using SUPER © video converter (by eRightSoft)

1) Drag and Drop the video you want to convert in Super

DROP A VALID MULTIMEDIA FILE HERE - Will try to play or encode
Active
FileName | Streaming Link
Output Status
70.208 MB
Encode (Active Files)
An ERROR has occurred. Click to read more...
Play (Active Files | Streams)
Preview The Last Rendered File
Player Options
Cancel Current
Cancel All

Figure 270

2) Select **AVI** from the **Output Container** list and select **H.264/AVC** from the **Output Video Codec**.

Figure 271

3) Click the**H264 Profile** button you can see on the right side of main window and from the opening dialog choose **BaselineProfile** and **Level 3**.

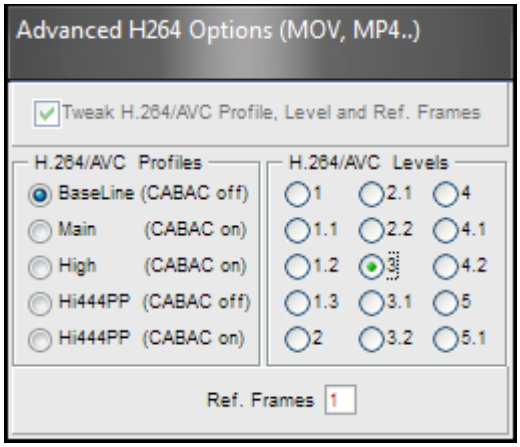


Figure 272

4) Enable the checkbox **Disable Audio**

5) Click **Encode (Active Files)** to start encoding of the videos.

Now you can open the videos with a standard video player (e.g. Windows Media Player) and check the quality. You can add the resulting video to the playlist of the Media Player Widget.

31.5.1 Using Media Player in JavaScript

The Media Player Widget can be also referenced in JavaScript programs with the following syntax:

```
//get the mediaplayer widget.
var mediaWgt = page.getWidget('MediaPlayerWgt2');

//load the play list
mediaWgt.setProperty('medialist', '/Storage Card/demo_3.avi,/Storage
Card/video1_3.avi');

// set the loopstyle 0 - noloop, 1 - loop one, 2- loop all, 3 - random
mediaWgt.setProperty('loopstyle', 2);

//start playing the first file.
mediaWgt.mediapath = '/Storage Card/demo_3.avi';
```

For a complete reference about JavaScript refer to chapter Working with **JavaScript**.

31.6 IPCamera Widget

An IPCamera widget is available in **Widget Gallery.** Using this widget is possible to show images captured from an IPCamera or show a video stream.

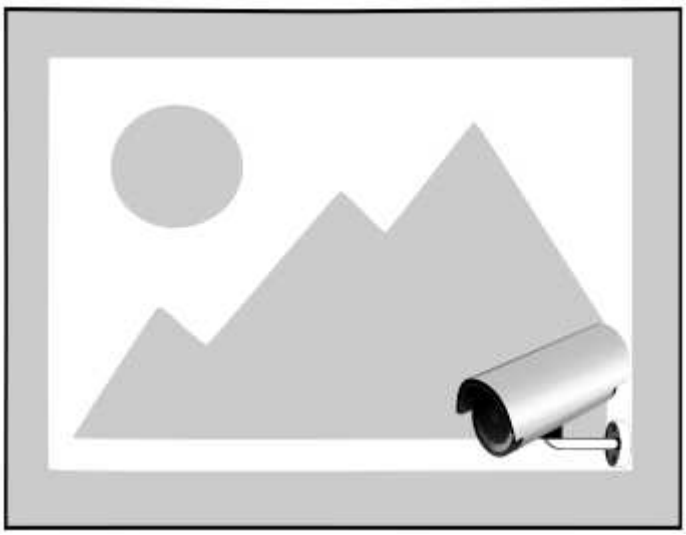


Figure 273

Follow the list of main parameters available for IPCamera widget:

Camera URL	URL of the IPCamera when used in JPEG format.
Refresh Rate	Number of JPEG images for second allowed. The max frame rate is 1fps.
User Name	Useful when IPCamera device is protected by a username & password.
Password	Useful when IPCamera device is protected by a username & password.
MJPEG Camera URL	IPCamera widget can be used also with streaming HTTP MJPEG. In this case parameters “Camera URL” and “Refresh Rate” are ignored. “MJPEG Camera URL” is the URL of MJPEG streaming. Ex. *http://192.168.0.1/video.cgi*

The only supported protocol is HTTP. For showing single frames the only supported format is JPEG while for streaming the only supported format is Motion JPEG.

Performance of streaming is not fixed and depends on many factors like: Frame size, Frame compression level, CPU of HMI Panel, Quality of IPCamera. Based on these factors the widget can reach upto 25 fps.

Multiple widgets are supported, however in this case performance could reduce framerate of each single widget.

The supported authentication methods are:

- Basic
- NTLM version 1
- Digest-MD5

No settings are required to select the method as it comes as an automatic selection from the camera web server to which the widget connects.

31.6.1 IPCamera tested

Follow the list of IPCamera tested at today.

IPCamera	Protocol	URL
D-Link DCS-932L	MJPEG	http://{ip_address}/video.cgi
Panasonic WV-Series Network Camera	MJPEG	http://{ip_address}/cgi-bin/mjpeg
D-Link DCS-5605 PTZ	MJPEG	http://{ip_address}/video/mjpg.cgi
D-Link DCS-900W IP Camera	MJPEG	http://{ip_address}/video.cgi
Hamlet HNIPCAM IP Camera	MJPEG HTTP	http://{ip_address}/video.cgi http://{ip_address}/image.jpg
DAHUA DH-IPC-HD2100P-080B 1.3mp Outdoor Vandalproof	HTTP	http://{ip_address}:9988/onvif/media_service/snapshot
Apexis APM-J901-Z-WS PTZ IP Camera	MJPEG HTTP	http://{ip_address}/videostream.cgi http://{ip_address}/snapshot.cgi
MOXA VPort 254 (Rugged 4-channel MJPEG/MPEG4 industrial video encoder)	MJPEG HTTP	http://{ip_address}/moxa-cgi/mjpeg.cgi http://{ip_address}/moxa-cgi/getSnapShot.cgi?chindex=1
NVS30 network video server	MJPEG HTTP	http://{ip_address}:8070/video.mjpeg http://{ip_address}/jpg/image.jpg
Foscam FI8916W	MJPEG HTTP	http://{ip_address}/videostream.cgi http://{ip_address}/snapshot.cgi

31.7 PTZ Controls

A pan–tilt–zoom camera (PTZ camera) is a camera that is capable of remote directional and zoom control. PTZ is an abbreviation for pan, tilt and zoom and reflects the movement options of the camera.

Widget *PTZ Controls* use action **MoveIPCamera** to send HTTP/cgi commands to the PTZ IPCamera. Following parameters ara available:

Camera URL URL of IPCamera. Ex. *http://192.168.10.123*

User Name	Useful when IPCamera device is protected by a username & password.
Password	Useful when IPCamera device is protected by a username & password.
Command	Command to send to PTZ controller. Ex. *decoder_control.cgi?command=0*

The supported authentication methods are:

- Basic
- NTLM version 1
- Digest-MD5

No settings are required to select the method as it comes as an automatic selection from the camera web server to which the widget connects.

31.8 Multistate Image Widget

Multistate Image is a widget designed to show an image from a collection based on the value of a tag used as Index. This widget may be used also for simple animations.

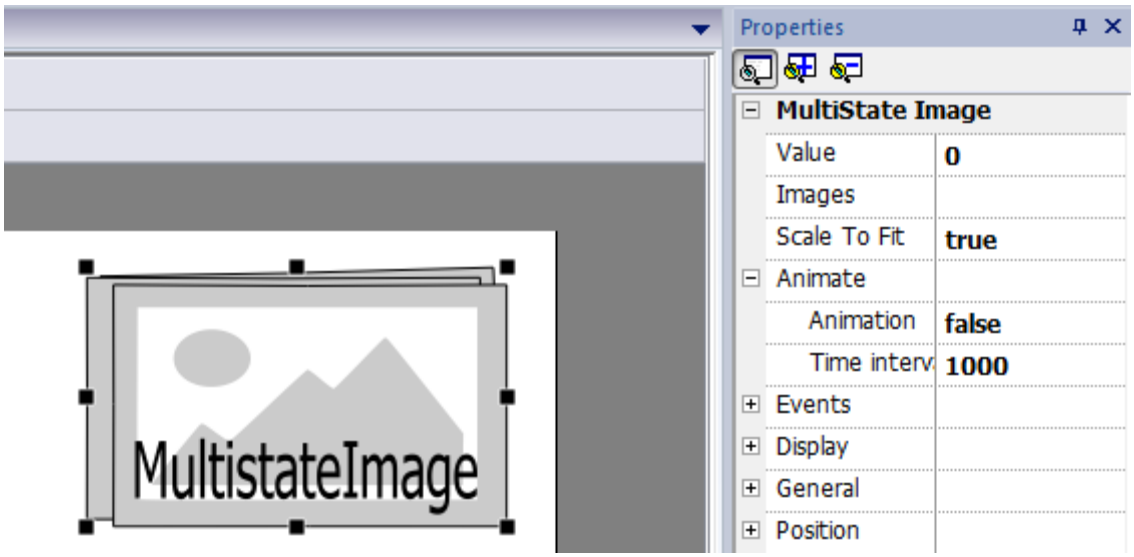


Figure 274

Follow the list of parameters of MultistateImage widget:

Value	Index of Image to show in Widget. By attaching this property to a tag it is possible to use this tag as Index, as a selector. Ex. value=0 means show Image with Index 0 in “Images” collection.
Images	Images collection. **Add/Remove Images** is used to add/remove images that later can be shown using related **Index**.
Animate	When **Animation** is true, a slideshow is shown and images change every 1000 ms (default **Time interval**).

31.9 Multistate Image Multilayer

The Multistate Image Multilayer widget extends the features available in Multistate Image widget. Multiple layer support allows user to define more features for animations and select the best at runtime based on different situations.

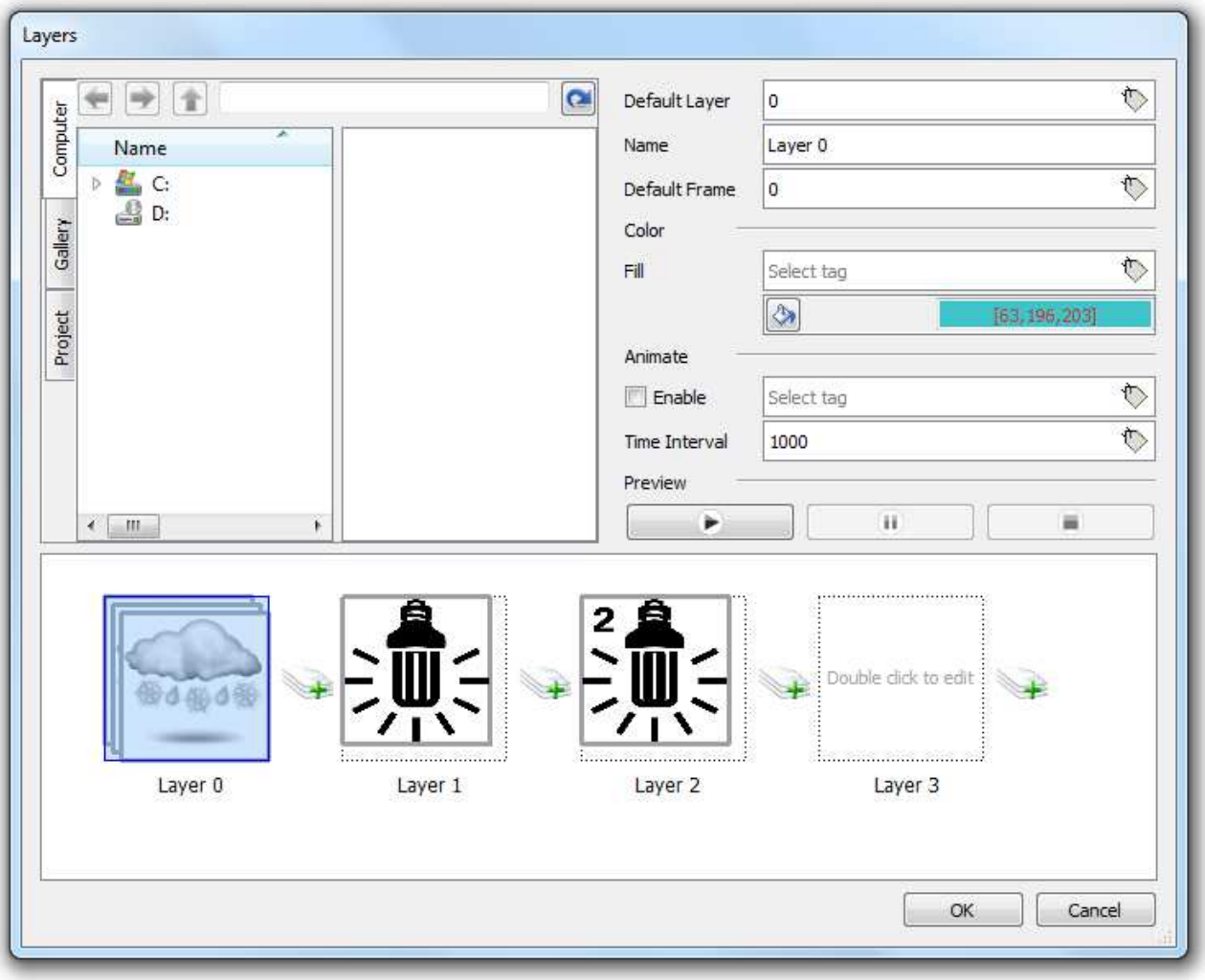


Figure 275

To use the widget proceed as follows:

1) Drag and Drop widget from widgets gallery into the page. Widget is available in category Basic -> Images.
2) Click on **Layers** to open configuration dialog
3) Define all layers needed. To add a new layer, use the **+** icon on the right of existing layers.

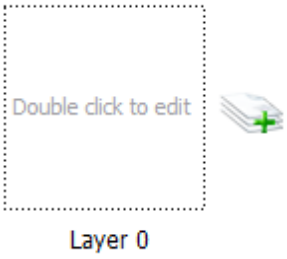


Figure 276

4) Double click on each layer to configure images that compose the layer.

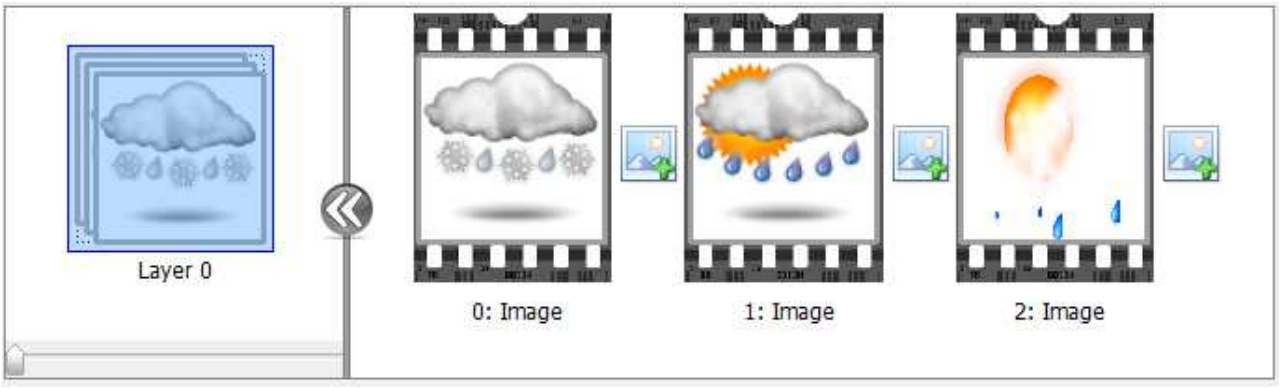


Figure 277

5) Browse images from the project, gallery or any folder in the computer; drag and drop images into the frame to add it to current layer. To add a new frame, use icon on the right of existing frames.

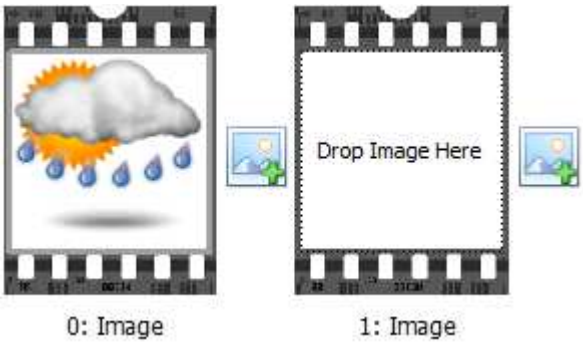


Figure 278

6) Customize widget using properties

Default layer	Default active layer shown at runtime when page is shown. Active layer can be changed at runtime, attaching it to a tag.
Name	Name of selected layer / current layer.
Default frame	Default frame shown when current layer is active. Active frame can be changed at runtime, attaching it to a tag.

Color / Fill	Fill color to use for images of current layer. Fill color can be changed at runtime, attaching it to a tag.
Animate	Enable/Disable animation of frames. When enabled, images of active layer change like in a slideshow. Animations can be start/stop at runtime attaching it to a tag.
Time Interval	Time interval of slideshow (ms). Used just when animation is enabled.
Preview	Preview is working like a simulator for animations. Can be used to verify animation in designing phase of widget without execute entire project.

31.10 Combo Box Widget

The Combo Box widget is available in the widget gallery and is already used by many widgets as a selector widget or as a way to filter rows shown in a table (like alarms or trends) based on values selected in the combo box.

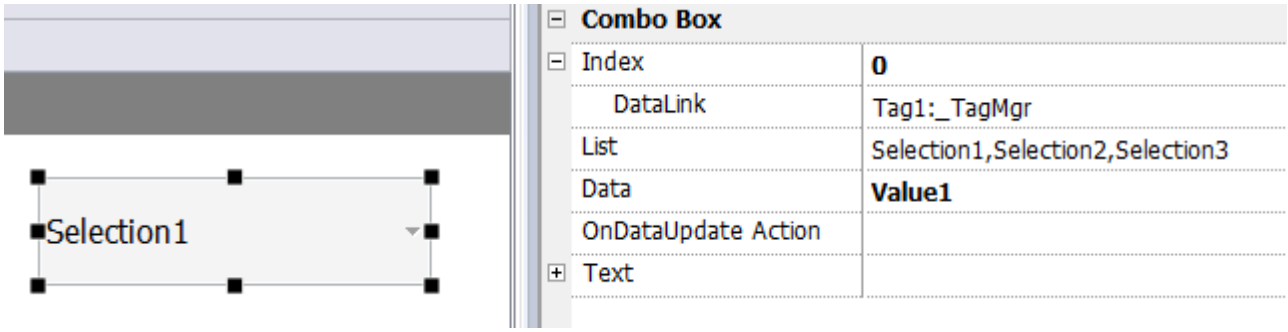


Figure 279

Follow the list of parameters of Combo widget:

Index	Each item listed in a combo has an index 0...n. This field returns the index (integer) of selected item in combo.
List / String List	Strings to show as items into combobox widget. This field is multilanguage.
Data / Data List	Optional parameter available in **advanced** mode that allows returning in the field **Data** of the widget, the related value reported in the **Data List** (as string). Usually **Index** is enough for 90% of applications, however sometime it is useful to return a custom value based on an item selected in the combo box.

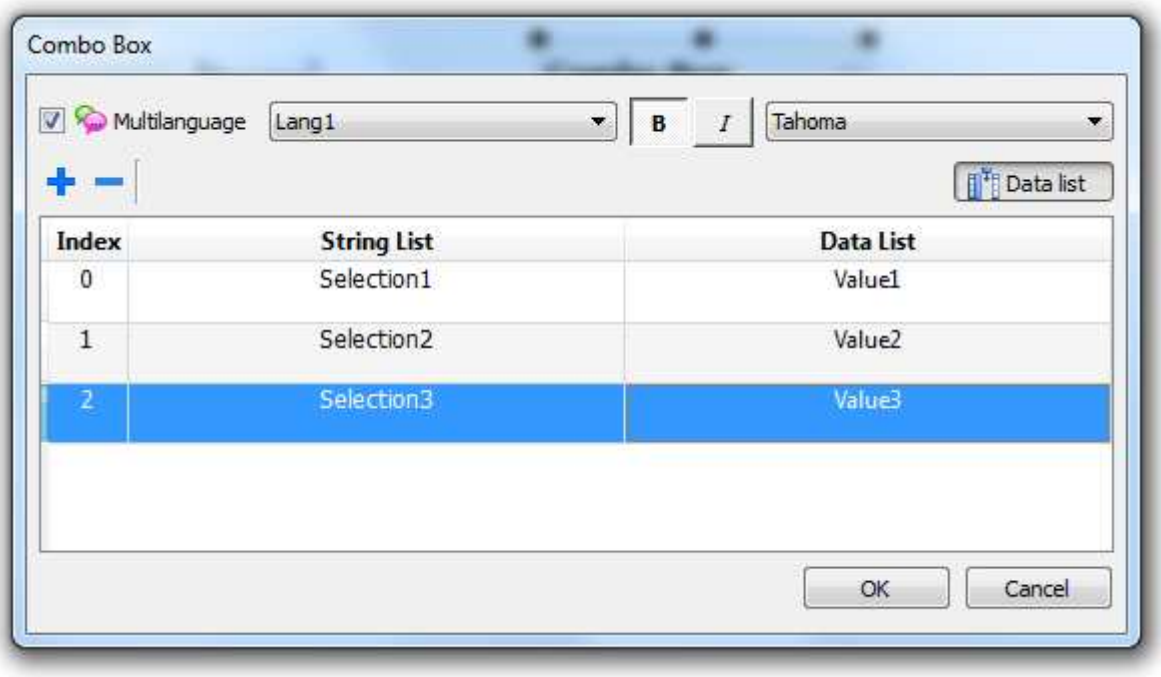


Figure 280

It is useful in many applications to attach fields like **Index** or **Data** to tags to know the values related to the selected item in the combo box. When attached to **Index**, the tag will contain the index (integer) of the selected item (0...n), when attached to **Data**, it will contain the data value (string) specified in the **Data List.**

31.11 Consumption Meter Widget

Consumption meter widget is available in the widgets gallery. This widget has been designed specifically to monitor a resource which is continuously increasing. The system reads the value of the resource and calculate the increment in a predefined range of time, the increment is then represented in a trend-like window, using bar-graphs. A typical application for this widget is the calculation of the power consumption of a system.

Representation is done using a bar graph where it is also possible (when range is weekly) to assign different colors based on the time frame.

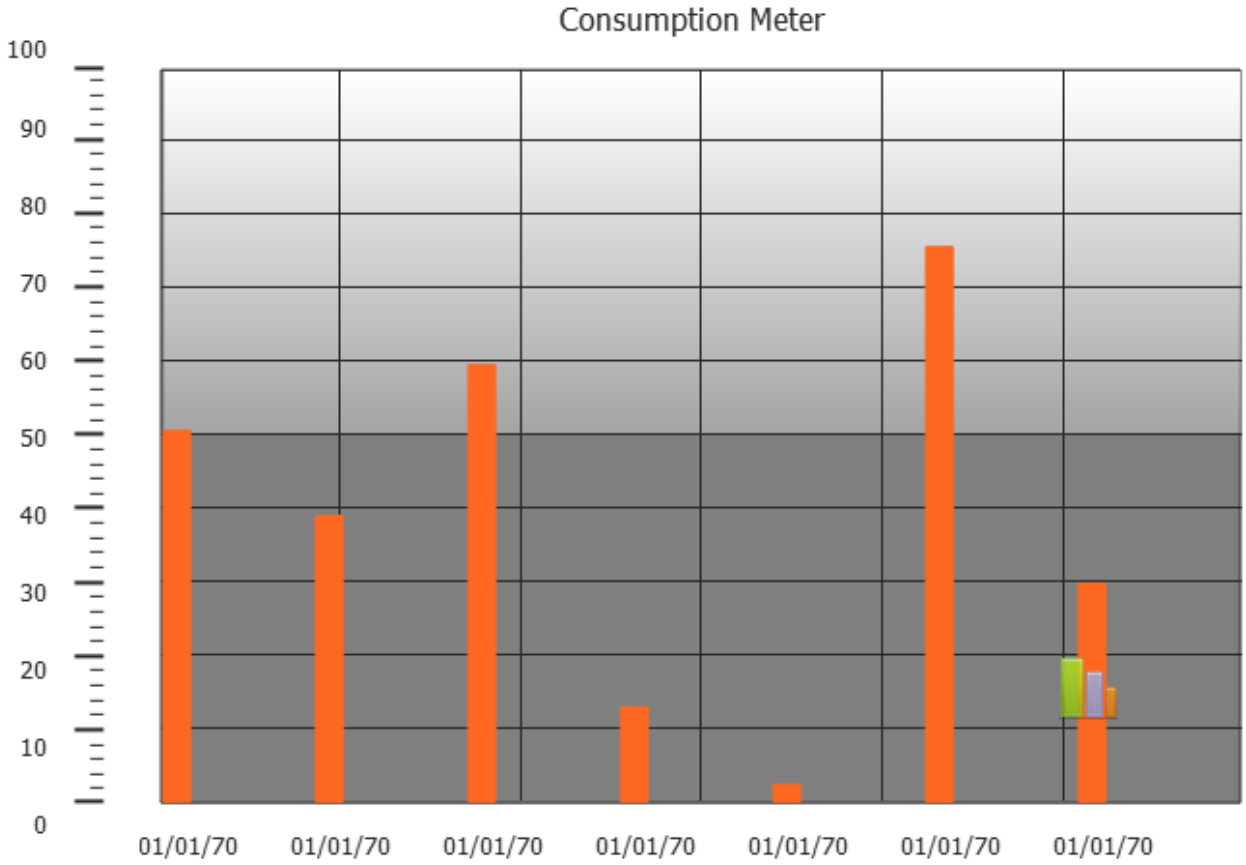


Figure 281

Below you can find a description of the main properties of this widget:

Value	this is the resource monitored by the system
Graph Duration / Graph Duration Units	these properties determines the time period that will be represented in the trend window
Bar Duration/Bar Duration Units	these properties determines the time period represented by each bar composing the graph
Time Periods	this property allows to highlight, with a minimum resolution of 1 hour, the increment of the monitored resource in a determined time period by using a specific color. Each bar composing the graph will then be represented using different colors, each showing the increment of the monitored resource in the corresponding time period.

Example: design a monitor for the consumption of energy with a weekly scale and a daily unit. Follow these steps to configure the widget:

1. Add protocol
2. Add Tag and link it to the physical variable to monitor (total energy consumed – ex. KW/h), we can call this Tag KWh. This tag contains an incremental number that summarize how many KW/h has been consumed from when energy started.
3. Add a Trend and link it to the KWh Tag to monitor

4. Add Consumption meter widget into a Page
5. Attach **Value** property of Consumption Meter to the KWh Tag.
6. Set **Graph Duration/Units** to 1 Week (range of time considered by Widget). This allow us to have a weekly graph of consumed energy.
7. Set Bar Duration/Units to 1 day (range of time where calculate consume of energy)
8. In Properties -> **Consumption Meter, you can** change the number of labels to show in the bar graph (ex. **X Labels** = 7 if we need a weekly graph).
9. **Open Time Periods** to access a configuration dialog that allow the setting of different colors for different values of the Tag KWh monitored in each bar.
10. Add as many color bands as you need, in this example we've added 3 color bands.
11. Assign to each hour in the weekly table the related band. In this example a red band (E1) was used to indicate the range of time in the day/week where the cost of energy is the highest.
12. For each band, if needed, a scale factor can be assigned.

The result is a consumption meter as a bargraph that shows daily consumption of energy (KW/h in this example) where the colors indicate the different energy cost that has been consumed. The height of each bar represents the amount of energy in the range of time considered (1 day in this example).

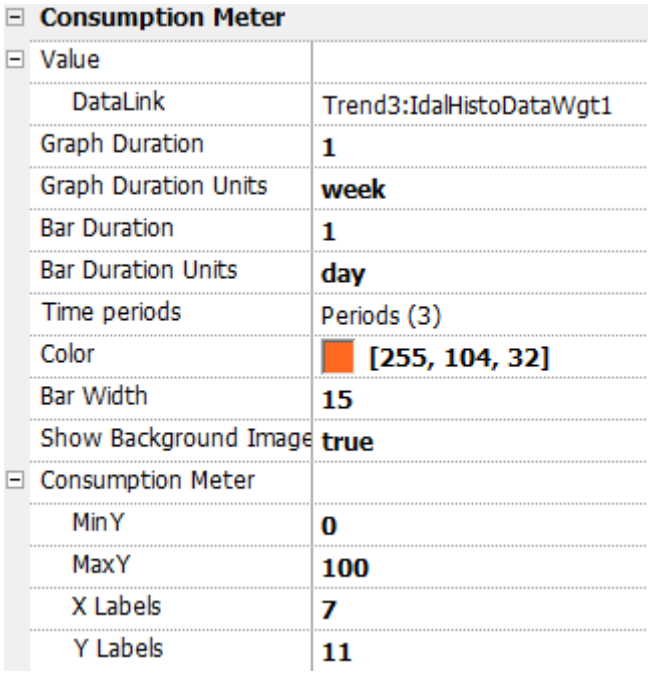


Figure 282

In **TimePeriods dialog**, to assign the color to the cells of the table, you can select the cells and click on the related band/color. Another way is to enter the index value of the band (1, 2, 3) into the cell to color it.

Figure 283

The Macro **ConsumptionMeterPageScroll** can be used to scroll the bar graph back and forth. The macro **RefreshTrend** is necessary to refresh the bar graph because it is not automatically refreshed. All other Macro of Trends are not supported today by the consumptionMeter.

31.12 Browser widget

An HTML5 compatible browser widget based on the WebKit engine has been added to the widget gallery.

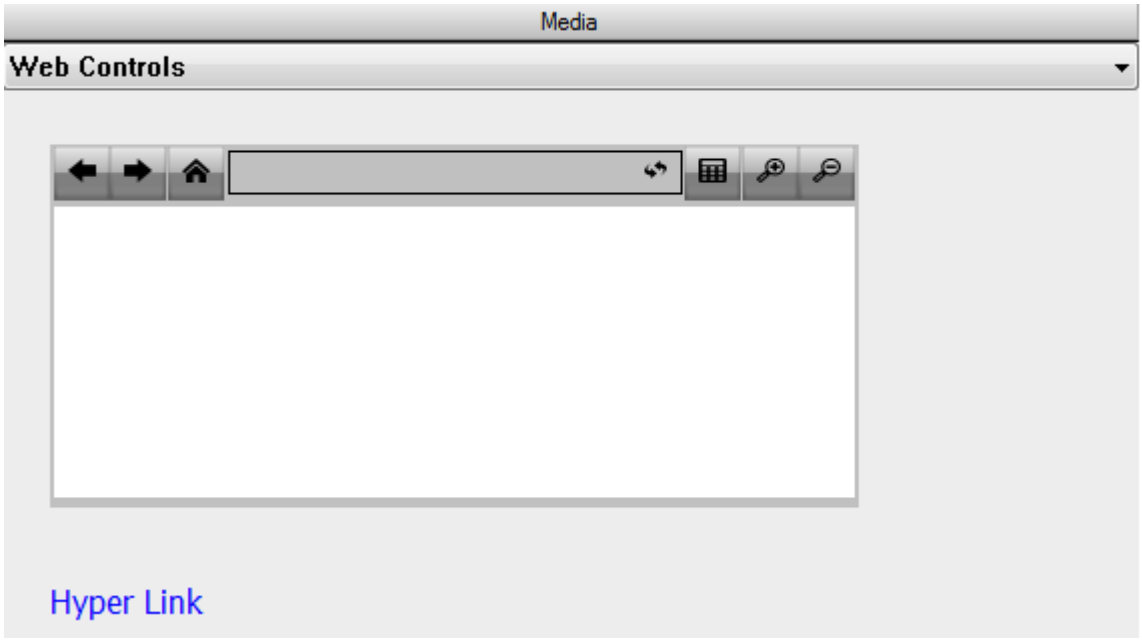


Figure 284

Browser widget can be used to embed web pages into hmi pages. Browser widget is not supported by MIPS based panels and ActiveX.

Widget properties allow user to customize many parts of the widget like:

- Home Page
- Scrollbars
- Zoom to fit option (enable/disable)
- Page load timeout (seconds)

For embedded hmi panels WCE based, the WebKit library (needed for using widget browser) has been designed as a plugin (ref. to Plugins chapter for more details) that developer can enable and download to the runtime only when required. This has been done to save around 3Mb of space required by this library when the widget is not used in the project.

Hyper Link widget is available to create labels into hmi pages working as

32 Custom Widgets

JMobile Studio has a large widget library which includes predefined dynamic widgets (such as buttons, lights, gauges, switches, Trends, Recipes, and dialog items), as well as static images (such as shapes, pipes, tanks, motors, etc.). With the widget library, you can simply drag and drop an object onto the page, and then size, move, rotate or transform it any way you want. All widgets in the gallery are vector based, so they look good at any size.

Custom widgets are widgets created by the user and based on the existing widgets from the gallery. This chapter describes how to create a custom widget and assign properties to it.

The advantage of the custom widget is that it can be built out of several elements, but with the flexibility to select only the necessary properties to be published and made available in the “custom widget" Property pane.

32.1 Creating a Custom Widget

The following steps describe how to create a custom widget:

1: Select all the Widgets you want as part of the custom Widget and group them.

2: Right click on the group, and select "Convert To Widget" from the context menu.

A Convert to Widget dialog box is shown below.

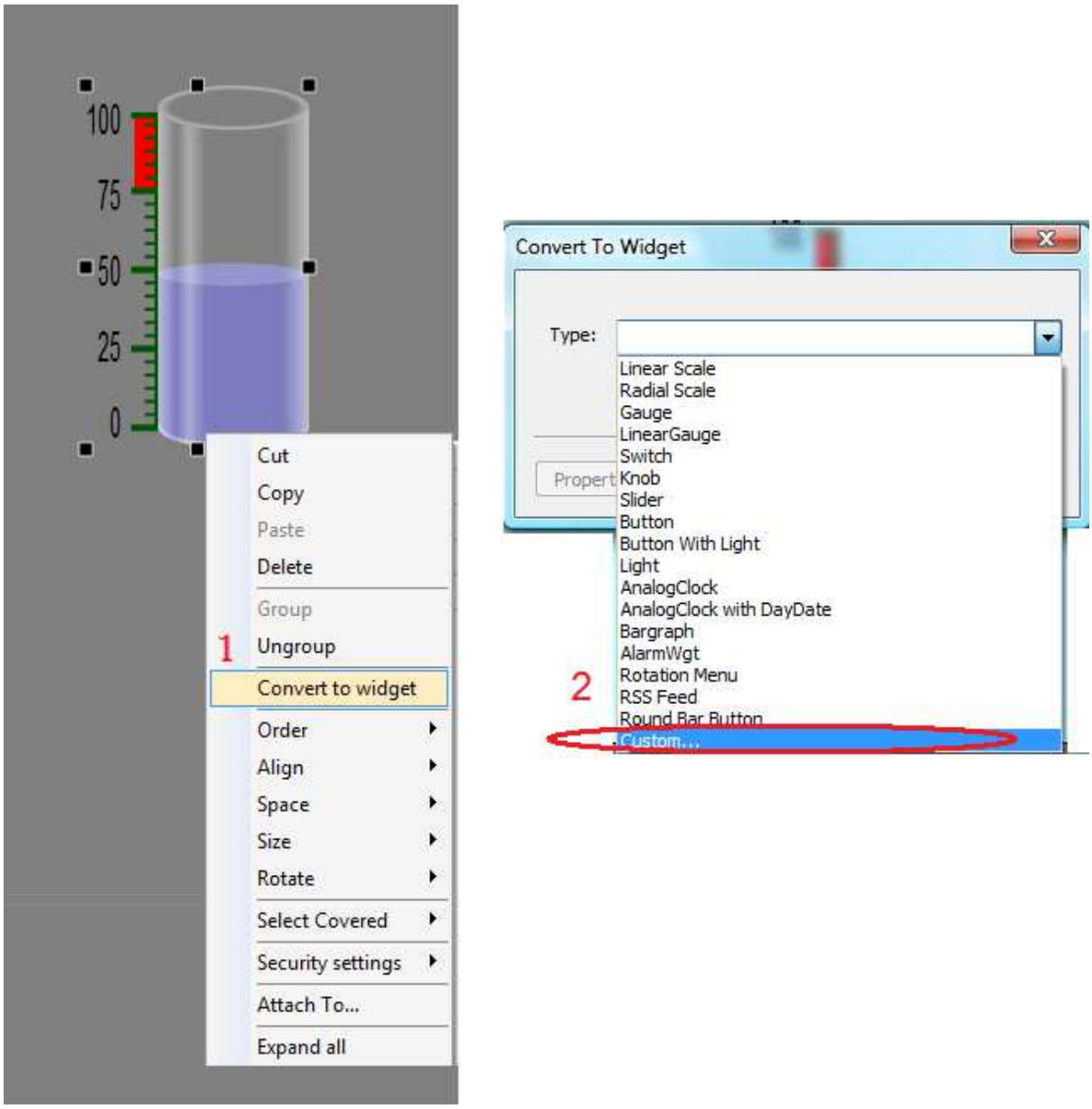


Figure 285

You can select existing custom widget types (such as Knobs, Button with Light, etc.), or you can select "Custom" to create a new custom widget type.

32.2 Adding the Properties

After creating the custom widget, the next step is to add the properties that will be published in the custom widget property pane. The “Property Select" dialog shows all the applicable properties for the grouped widget; this is basically a list of all the properties of the individual widgets grouped together. You can select the properties that you want to expose for that custom widget by clicking the corresponding check box.

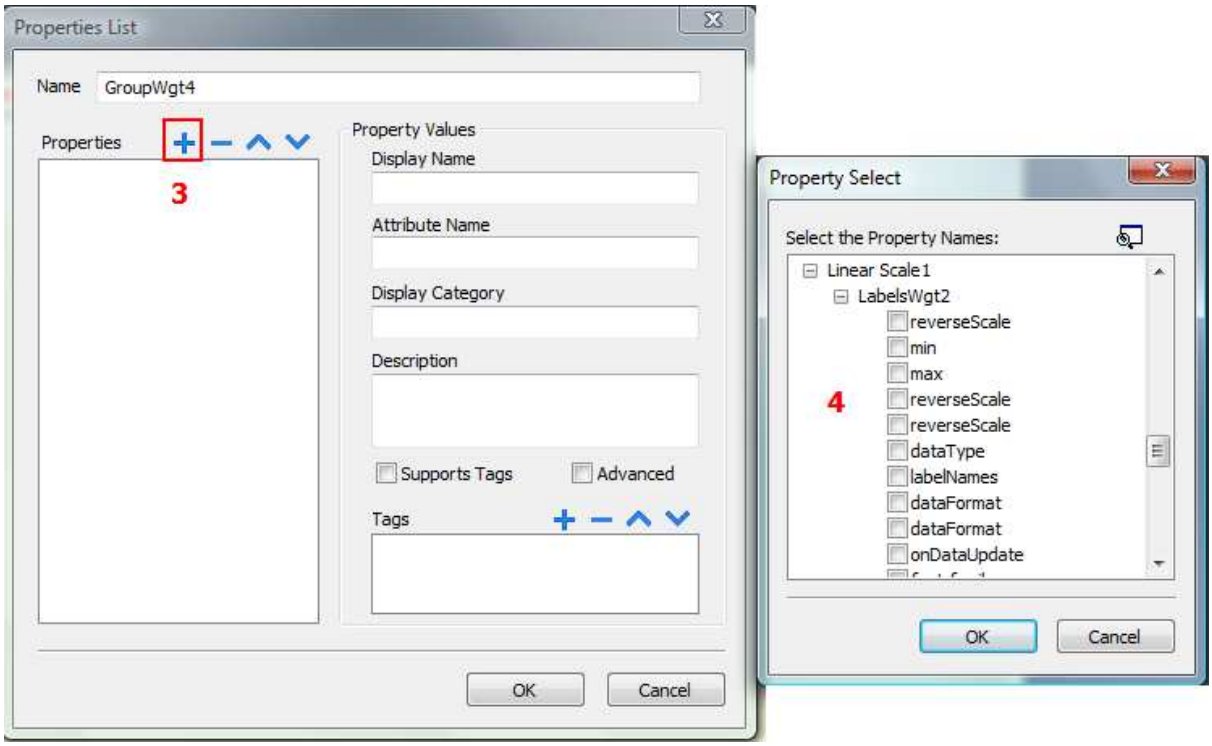


Figure 286

Enter the name of the custom widget. This is the name that will appear in the Property view.
The next step is to select the properties that will be displayed in the Property view. Click on the '+' button above the 'Properties' list box, and a Property Select dialog will be displayed.

NOTE *The ConvertToWidget dialog shows "standard" custom widget types. These types are defined in the gallery. The dialog, however, does not show types that are specifically created for a project.*

Display Name
Name that will be shown in the Property view. You can change it to set the information for each custom widget property.

Attribute Name
The name exposed by JMobile Studio, to JavaScript functions and Attach Tag dialog. The default property name has the form 'WidgetType.name'; 'WidgetType' is the type of widget; and 'name' is the attribute name. If you have more than one widget of the same type, the widget type name will be WidgetType01, WidgetType02, etc.

Display Category
The category or group of the property in the Property view. All properties in the same category are shown together in the Property view. This allows you to organize the properties in the view. The Display Category allows you to view by category group, by clicking on either the Collapse or Expand button. For example, you can declare position properties, like the X coordinate, height, width properties in a single display category called Position.

Description
Allows you to define the description and comments within the property; the information will be displayed in the property pane.

Advanced
The properties are shown in either the "Normal" or "Advanced" mode. The "Advanced" check box allows you to specify whether each property should appear in the advanced, or in the simple property pane view mode.

Support Tags
The “Supports Tags" checkbox must be marked if the property supports the “Attach to" attribute.

Tags
The “Tags" list box indicates the internal Tag name for the Widget. This internal Tag name is typically the same as the attribute name; however, this is not always the case. You can assign a different attribute name for your custom Widget. The Tag list is also used to combine Tags.

If you want to combine two or more properties into one, select the primary property in the Property List and click on the ‘+’ button above the Tags list box. The Property Select dialog will be displayed, and you can select the properties that should be combined. Note that this dialog box only shows the properties that should be combined (not all properties are shown in the Properties list). For example, to combine the 'min' property of the scale Widget and Bar graph Widget, click on the NeedleWgt min property and click on the BargraphWgt min property from the Property Select dialog. Click the OK button. Both attributes will be shown in the Tags list box, as shown in the figure below.

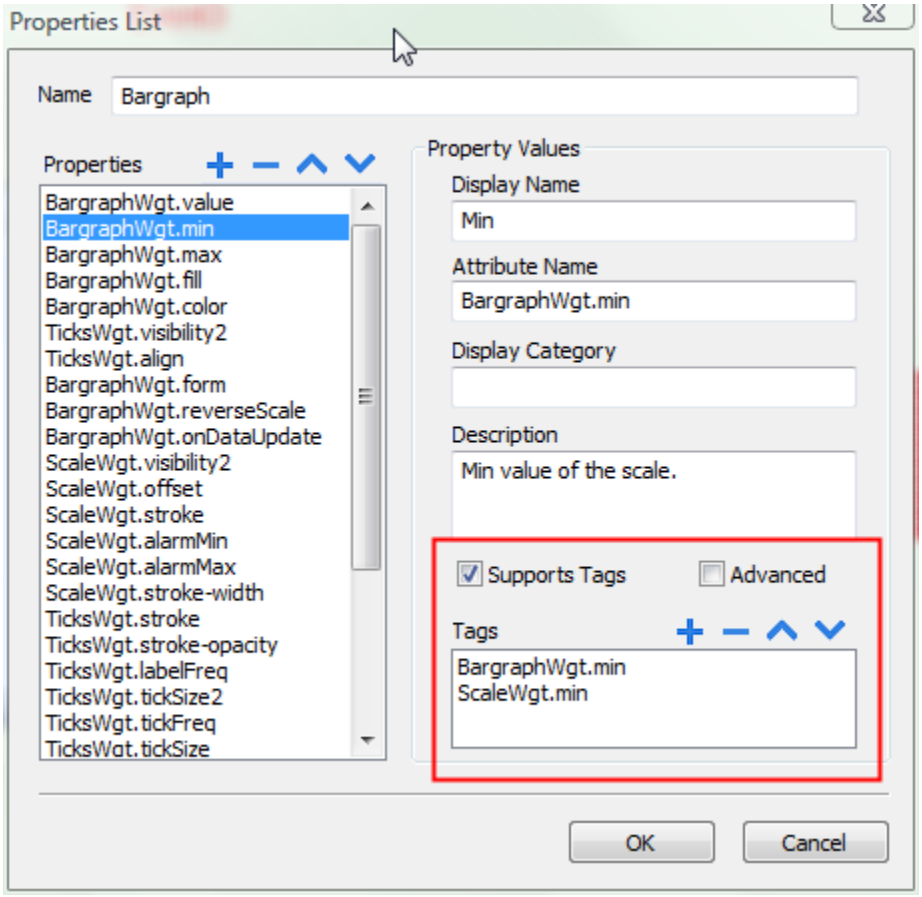


Figure 287

You can arrange the order of the properties by clicking on the up or down button in the Property List. To remove a property, select the property name and click on the delete button.
When a property is selected in the Property List, the property information is shown in the dialog box.

NOTE *Custom widgets usually are composed by many sub widgets. For example a button is a complex widget composed by two Image widgets, a button widget and label. This is clearly visibile in the* ***ObjectView*** *when the widget is selected. To select a sub widget like the label in a button, use ObjectView or* ***Shift + leftClick*** *of mouse. In this way sub widget can be changed without ungroup all widget.*

32.3 Editing Custom Properties

If you want to change the properties of a Custom Widget after it has been created, you can simply right click on the Widget in the Page editor and select the "Custom Properties" menu item from the context menu. The Custom Properties dialog will be displayed and you can change the properties.

33 Send an E-mail Message

SendMail action can be used to send emails. You can include tags in the e-mail body and attachments.

The *SendMail* action has been design for working with alarms and schedulers but can be executed as conseguence of many other events.

NOTE *The system does not yet support SSL/TSL.*

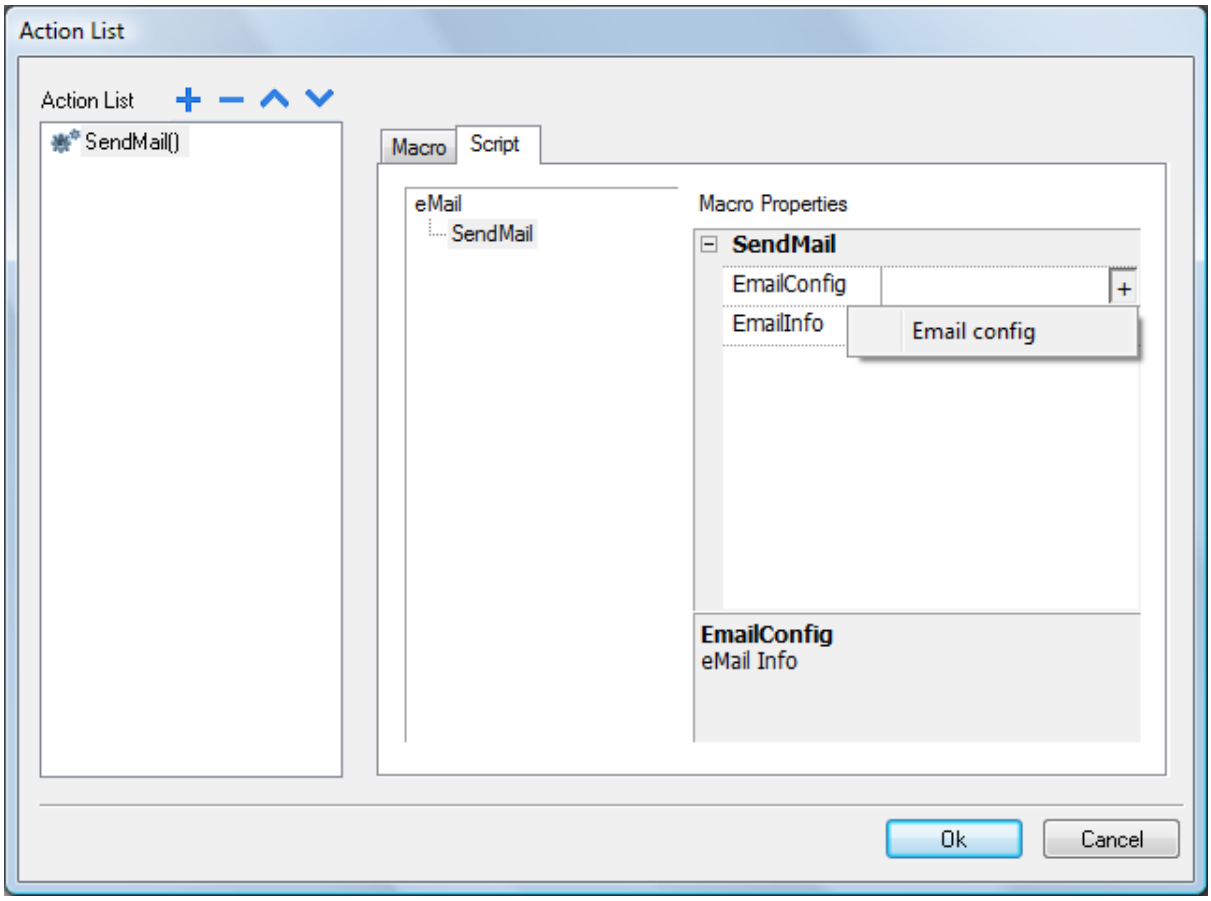


Figure 288

33.1 Configure E-mail Server

To configure the email server, you need to provide the following information into field ***EmailConfig*** of SendMail action:

SMTP SMTP server address

Server Name optional – it can be used for information purposes

Server Port Port to use to connect to SMTP server. Default is 25.

Require Auth Flag it if SMTP server requires authentication

User Name Username to use for sending mail using SMTP server

Password Password to use for sending mails using SMTP server

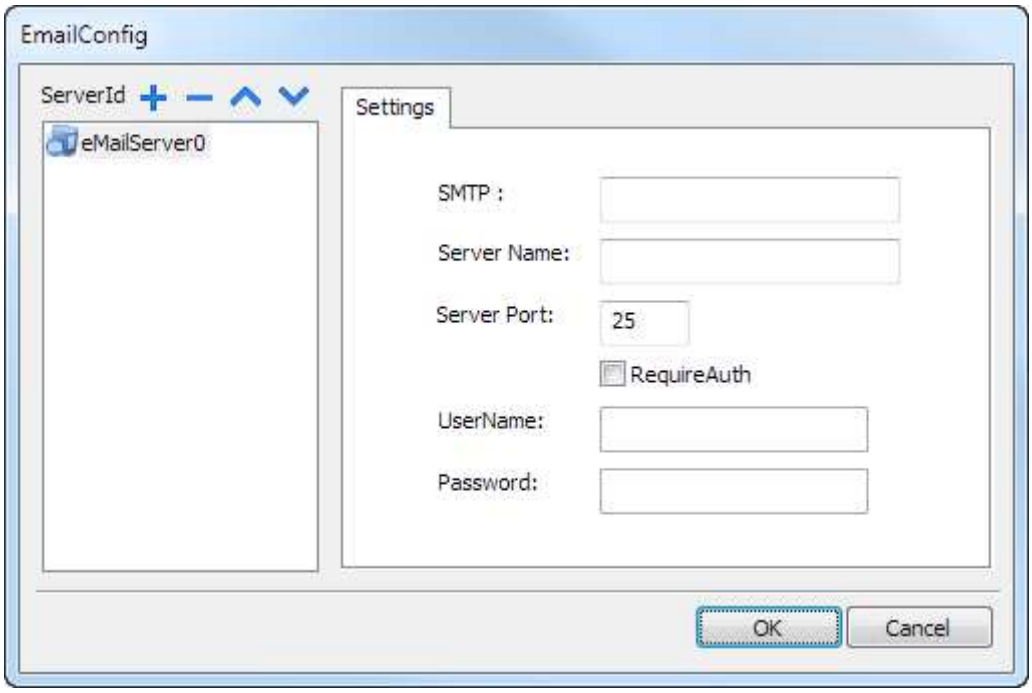


Figure 289

NOTE You can add more E-mail servers by clicking the “+" button on the left hand side.

33.2 Configure E-mails

In the ***emailInfo***, configure emails using following parameters:

Name optional – it can be used for information purposes

Description optional – it can be used for information purposes

From Sender email address (ex. John@domain.com).

To Recipient e-mail addresses. If you want to send the email to more than one recipient, separate the e-mail addresses with a semi-colon “;"

Subject Subject of email

Attachment Path of the file to send in attach to the email. The system support one single attachment for each email sent.

NOTE *SMTP servers usually limit the max size allowed for an E-Mail attachment.*

Body Main content of the email. System support **Live Tags**.

You can add more emails templates by clicking the “+" button on the left hand side.

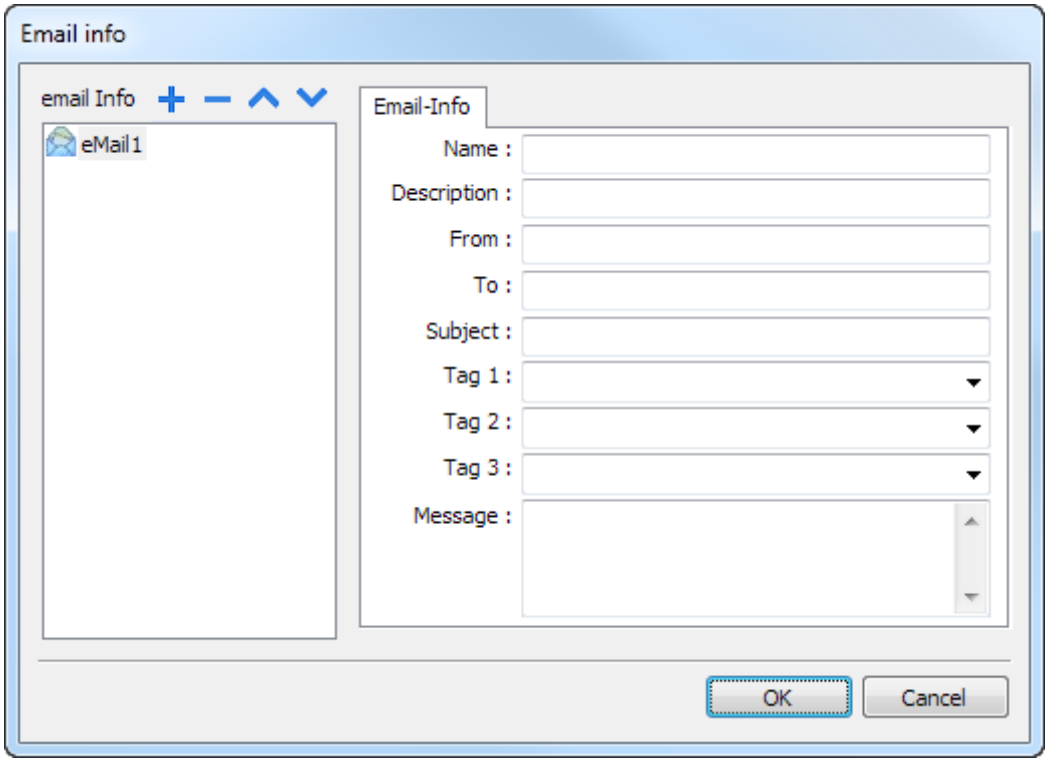


Figure 290

34 JavaScript

The purpose of this chapter is to describe the JavaScript interface implemented in JMobile Studio. JMobile Studio JavaScript is based on the ECMAScript programming language http://www.ecmascript.org , as defined in standard ECMA-262. Microsoft Chakra and Firefox SpiderMonkey JavaScript engines support the ECMAScript standard. If you are familiar with JavaScript, you can use the same type of commands in JMobile Studio as you do in a web browser. If you are not familiar with the ECMAScript language, there are several existing tutorials and books that cover this subject, such as:

https://developer.mozilla.org/en/JavaScript

This purpose of this document is not to explain JavaScript language, but rather to describe how JavaScript is used in the JMobile Studio applications.

34.1 Execution

A JavaScript function is executed when an event occurs. For example, a user can define a script for the **`OnMousePress`** event and the JavaScript script will be executed when the button is pressed on the panel. It is important to note that JavaScript functions are not executed in the same manner as certain other controller programming scripts, such as Ladder Logic. JavaScript functions are not executed at a given scan rate the whole time, but they are only executed when the given event occurs. This approach minimizes the overhead required to execute logic on the panel.

JMobile Studio provides a JavaScript engine running at the client side. Each project page can contain scripts with scope local to the page where they are programmed. The project can also contain global scripts that can be executed by scheduler events or alarm events, but it is important to understand that the scripts are still executed at the client side. In other words, having more than one client connected to the panel (for instance, an external PC running the JMobile Client) means each client will run the same script, providing output results that depend on the input. Inputs provided to the different clients may be different.
This can be clarified, for instance, considering a situation in which the script acts based on a slider position, which can be different for the different clients.

34.2 Events

You can add the JavaScript in the following events:

- Widget Events
- Page Events
- System Events

For events of type *OnMousePress*, *OnMouseRelease*, *OnMouseClick* and *OnWheel*, JS *eventinfo* parameter contains the following additional properties:

eventInfo.posX local mouse/touch X coordinate w.r.t. widget coordinates

eventInfo.posY local mouse/touch Y coordinate w.r.t. widget coordinates

eventInfo.pagePosX page X mouse/touch coordinate

eventInfo.pagePosY page Y mouse/touch coordinate

eventInfo.wheelDelta mouse wheel delta. It is an integer value with sign. Sign represents the rotation direction. The actual value is the rotation amount in eighths of a degree. The smallest value depends on the mouse resolution. Typically this is 120 (corresponding to 15 degrees).

34.2.1 Widget Events

34.2.1.1 onMouseClick

```
void onMouseClick (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed and released quickly.

Parameters

me The object that triggers the event.
eventInfo Details of the event triggered.

```
function buttonStd1_onMouseClick(me, eventInfo) {
     //do something…
}
```

34.2.1.2 onMouseHold

```
void onMouseHold (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed and released after n seconds where n=**Hold Time** seconds specify in widget properties.

Parameters

me The object that triggers the event.
eventInfo Details of the event triggered.

```
function buttonStd1_onMouseHold(me, eventInfo) {
     //do something…
}
```

34.2.1.3 onMousePress

```
void onMousePress(me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed.

Parameters

me The object that triggers the event.
eventInfo Details of the event triggered.

```
function buttonStd1_onMousePress(me, eventInfo) {
     //do something…
}
```

34.2.1.4 onMouseRelease

```
void onMouseRelease (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is released.

Parameters

me	The object that triggers the event.
eventInfo	Details of the event triggered.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
     //do something…
}
```

34.2.1.5 onDataUpdate

```
boolean onDataUpdate (me, eventInfo)
```

This occurs when the data attached to the Widget changes.

Parameters

me	The object that triggers the event.
eventInfo	An object with these fields (you can refer fields using "." - dot notation):
oldValue:	The old value that is the widget value before the change.
newValue:	The new value that is the value which will be updated to the widget.
attrName:	The attribute on which the event is generated
index:	An integer attribute index if any, default = 0
mode:	W when user is writing to the widget, R otherwise.

This event is triggered by the system before the value is passed to the Widget; this means the code programmed here can modify or alter the value before it is actually passed to the Widget.
The code can terminate with a **return true** or **return false**.
After terminating the code with return false, the control is returned to the calling Widget that may launch other actions.
After terminating the code with true, the control is NOT returned to the Widget and this makes sure that no additional actions are executed following the calling event.

```
function buttonStd1_onDataUpdate(me, eventInfo) {
if ( eventInfo.oldValue < 0) {
     //do something…
}
     return false;
}
```

34.2.2 Page Events

34.2.2.1 onActivate

```
void onActivate( me, eventInfo )
```

This event occurs each time the page is shown.

Parameters

me	The object that triggers the event.
eventInfo	It is reserved for future enhancements.

This JavaScript will execute when the page is Active. It means that, when the page is loaded, the script will execute.

```
function Page1_onActivate(me, eventInfo) {
     //do something…
}
```

34.2.2.2 onDeactivate

```
void onDeactivate( me, eventInfo )
```

This occurs when leaving the page.

Parameters

me	The object that triggers the event.
eventInfo	It is reserved for future enhancements.

```
function Page1_onDeactivate(me, eventInfo) {
      //do something…
}
```

34.2.2.3 onWheel

```
void onMouseWheelClock( me, eventInfo )
```

This occurs when a wheel device is moving (ex. Mouse wheel).

Parameters

me	The object that triggers the event.
eventInfo	Details of the event triggered.

```
function Page1_onMouseWheelClock(me, eventInfo) {
      //do something…
}
```

34.2.3 System Events

There are three types of system events:

- related to the scheduler
- related to the alarms
- related to Wheel device

Be sure to use unequivocal function names in JavaScript between code at page & project level (scheduler/alarms).
When a conflict happens at runtime (two functions with the same name in current page and at project level), the system execute JavaScript callback at page level (not a project level).

When a JavaScript callback is not found in current page, the system search for it at project level automatically.

34.2.3.1 Scheduler Event

The event occurs when triggered by the proper action available in the scheduler system as shown in the figure below.

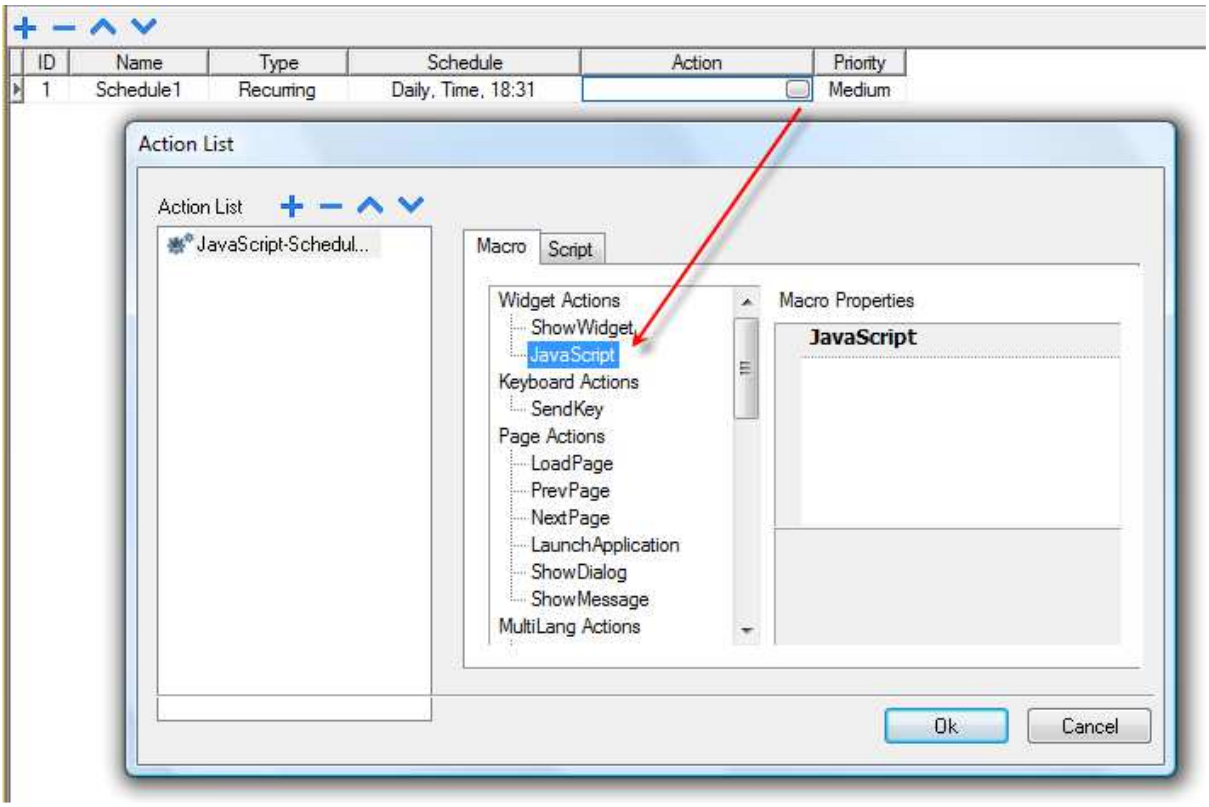


Figure 291

34.2.3.2 Alarm Event

The event occurs when triggered by a specific alarm condition and programmed in the proper action as shown in the figure below.

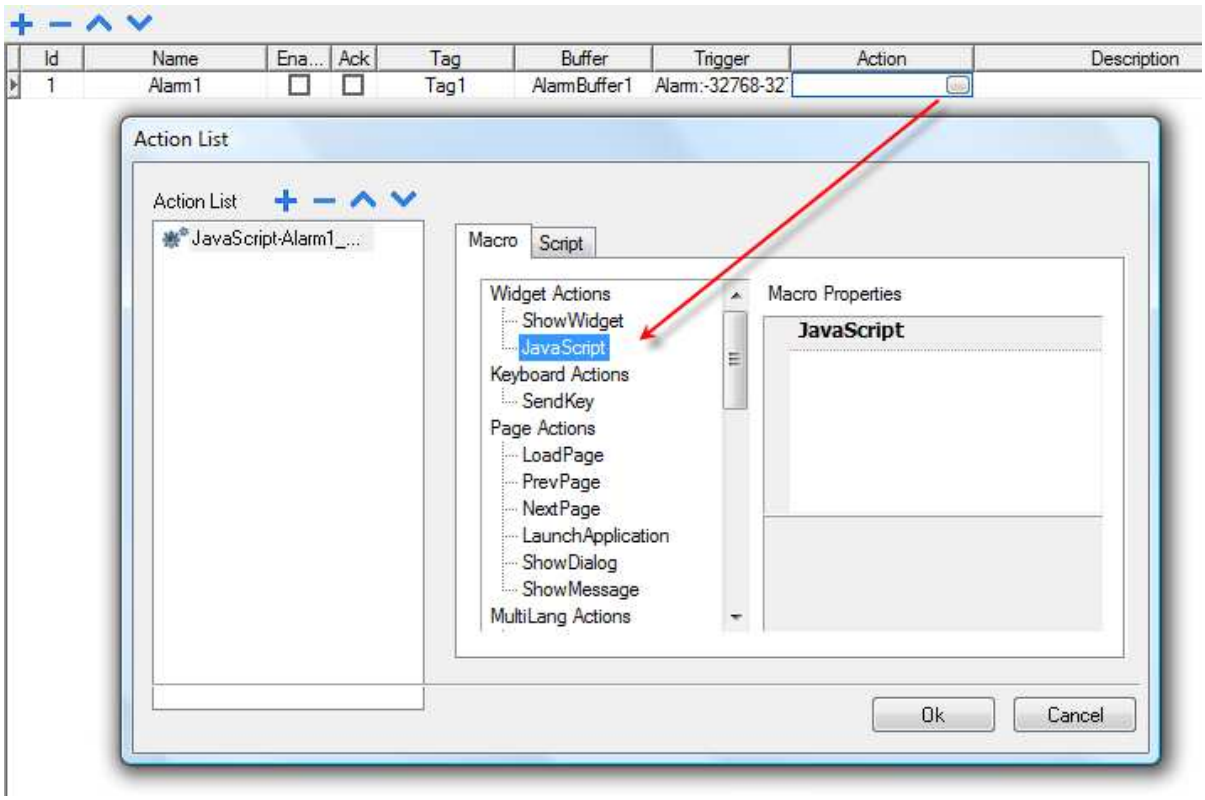


Figure 292

Once the system events are configured, the custom code for them can be edited from the global JavaScript editor interface, which is available from the Project view (double click on the project name icon) as shown in the figure below.

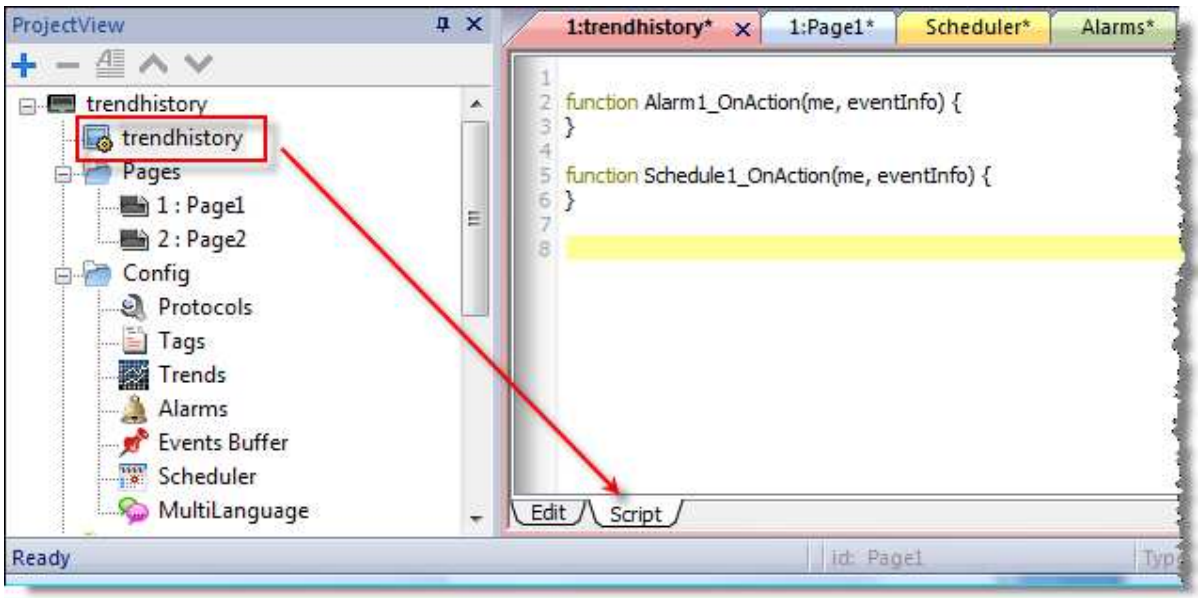


Figure 293

34.2.3.3 onWheel

```
void onMouseWheelClock( me, eventInfo )
```

This occurs when a wheel device is moving (ex. Mouse wheel).

Parameters

me	The object that triggers the event.
eventInfo	Details of the event triggered.

```
function Project1_onMouseWheelClock(me, eventInfo) {
      //do something…
```

34.3 Objects

JMobile Studio uses JavaScript objects to access the elements of the page. Each object is composed of properties and methods that are used to define the operation and appearance of the page element. The following objects are used to interact with elements of the HMI page:

Widget	The `Widget` class is the base class for all elements on the page including the page element
Page	This object references the current HMI page. The page is the top-level object of the screen
Group	A group is a basic logical element that is associated with a set of logical tags. It provides an interface to enable the uniform operation on a set of logically connected tags
Project	This object defines the project widget. The `project` widget is used to retrieve data about the project such as tags, alarms, recipes, schedules, tags and so on. There is only one widget for the project and it can be referenced through the `project` variable
State	Class for holding state of a variable acquired from the controlled environment. Beside value itself, it contains the timestamp indicating when the value is collected together with flags marking quality of the value.

34.3.1 Widget

The `Widget` class is the base class for all elements on the page including the page element. `Widget` is not a specific element but a JavaScript class.

IMPORTANT When you change the properties of widgets with JavaScript you have to set the widget **Static Optimization** to **Dynamic**, otherwise changes to properties will be ignored. You can find the option **Static Optimization** in the **Advance Properties**. Whenever a call to getWidget fails, remote debugger report following error:
"*Trying to access static optimized widget "label1". Disable widget static optimization to access widget from script.*".
This error is visible also using following:

```
var wgt;
try {
wgt = page.getWidget('label1');
```

```
} catch(err) {
alert("" + err);
}
```

The following properties are common among all widgets:

objectName
`string objectName`

It gets the name of the Widget. The name is a unique id for the Widget.

```
function btnStd04_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      var name = wgt.objectName;
}
```

x

`number x`

It gets or sets the Widget 'x' position in pixels.

```
function btnStd1_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.x = 10;
}
```

y

`number y`

It gets or sets the Widget 'y' position in pixels.

```
function btnStd1_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.y = 10;
}
```

width

`number width`

It gets or sets the Widget width in pixels.

```
function btnStd1_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.width = 10;
}
```

height

`number height`

It gets or sets the Widget height in pixels.

```
function btnStd1_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.height = 10;
}
```

visible

`boolean visible`

It gets or sets the Widget visible state.

```
function btnStd4_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.visible = false;
}

function btnStd5_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.visible = true;
}
```

value

`number value`

It gets or sets the Widget value.

```
function btnStd6_onMouseRelease(me) {
      var wgt = page.getWidget("field1");
      wgt.value = 100;
}
```

opacity

`number opacity (range from 0 to 1)`

It gets or sets the Widget opacity. Values are decimals from 0 to 1, where 1 is 100% opaque.

```
function btnStd8_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.opacity = 0.5;
}
```

rotation

`number rotation (in degrees)`

It gets or sets the rotation angle for the Widget. The rotation is done by degree and makes a clockwise rotation, starting at the East position.

```
function btnStd9_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.rotation = 45;
}
```

userValue

`string userValue`

It gets or sets a user-defined value for the Widget. This field can be used by JavaScript functions to store additional data with the Widget.

```
function btnStd9_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      wgt.userValue = "Here I can store custom data";
}
```

Every widget has some specific properties that you can access using dot notation. For an up-to-date and detailed list of properties you can use the Qt Script Debugger inspecting the widget methods and properties.

The following methods are common among all widgets:

getProperty

`object getProperty( propertyName, [index] )`

Returns a property

Parameters

`propertyName`	A string containing the name of property to get.
`index`	The index of the element to get from the array. Default is 0.

Almost all properties that are shown in the JMobile Studio Property view can be retrieved from the `getProperty` method. The `index` value is optional and only used for Widgets that support arrays.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
      var shape = page.getWidget("rect2");
      var y_position = shape.getProperty("y");
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
      var image = page.getWidget("multistate1");
      var image3 = image.getProperty("imageList", 2);
      //…
}
```

setProperty

`boolean setProperty( propertyName, value, [index] )`

Sets a property for the Widget

Parameters

`propertyName`	A string containing the name of property to set.
`value`	A string containing the value to set the property.
`index`	The index of the element to set in the array. Default is 0.

Almost all properties that are shown in the JMobile Studio Property view can be set by this method. The `index` value is optional and only used for Widgets that support arrays (for example a MultiState Image widget). The `setProperty` method returns a boolean value `true` or `false` to indicate if the property was set or not.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
      var setting_result = shape.setProperty("y", 128);
      if (setting_result)
            alert("Shape returned to start position");
```

```
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
      var image = page.getWidget("multistate1");
      var result = image.setProperty("imageList", "Fract004.png", 2);
      //…
}
```

34.3.2 Page

This object references the current HMI page. The page is the top-level object of the screen.

Follow the list of Page Object Properties:

backgroundColor

`string backgroundColor (in format rgb(xxx, xxx, xxx) where xxx range from 0 to 255)`

The page background color

```
function btnStd11_onMouseRelease(me) {
      page.backgroundColor = "rgb(128,0,0)";
}
```

width

`number width`

The Page width in pixels

```
function btnStd05_onMouseRelease(me) {
      var middle_x = page.width / 2;
}
```

height

`number height`

The Page height in pixels

```
function btnStd05_onMouseRelease(me) {
      var middle_y = page.height / 2;
}
```

userValue

`string userValue`

It gets or sets a user-defined value for the Widget. This field can be used by JavaScript functions to store additional data with the page.

```
function btnStd9_onMouseRelease(me) {
      page.userValue = "Here I can store custom data";
}
```

Follow the list of Page Object methods:

getWidget

`object getWidget( wgtName )`

It returns the Widget with the given name.

Parameters

`wgtName`	A string containing the name of widget

Return value

An object representing the widget. If the widget does not exist, `null` is returned.

```
function btnStd1_onMouseRelease(me) {
      var my_button = page.getWidget("btnStd1");
}
```

setTimeout

`number setTimeout( functionName, delay )`

It starts a timer to execute a given function after a given delay once.

Parameters

`functionName`	A string containing the name of function to call.
`delay`	The delay in milliseconds.

Return value

It returns a number corresponding to the timerID.

```
var duration = 3000;
var myTimer = page.setTimeout("innerChangeWidth()", duration);
```

clearTimeout

`void clearTimeout( timerID )`

It stops and clear the timeout timer with the given timer.

Parameters

`timerID`	The timer to be cleared and stopped.

```
var duration = 3000;
var myTimer = page.setTimeout("innerChangeWidth()", duration);
// do something
page.clearTimeout(myTimer);
```

setInterval

`number setInterval( functionName, interval )`

It starts a timer that executes the given function at the given interval.

Parameters

`functionName`	A string containing the name of function to call.
`interval`	The interval in milliseconds.

Return value
It returns a number corresponding to the timerID

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
```

clearInterval

`void clearInterval( timerID )`

It stops and clears the interval timer with the given timer.

Parameters
`timerID` The timer to be cleared and stopped.

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
// do something
page.clearInterval(myTimer);
```

clearAllTimeouts

`void clearAllTimeouts()`

It clears all the timers started.

```
page.clearAllTimeouts();
```

34.3.3 Group

A group is a basic logical element that is associated with a set of logical tags. It provides an interface to enable the uniform operation on a set of logically connected tags.

Follow the list of Methods supported by Group Object:

getTag

`object getTag( TagName )`

Gets the tag specified by `TagName` from the group object.

Parameters
`TagName` A string representing the tag name.

Return value
An object that is the value of the tag or if tag value is an array it returns the complete array. If you need to retrieve an element of the array, check the method `getTag` available in object `Project`. `undefined` is returned if tag is invalid.

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getTag("Tag1");
```

getCount

`number getCount()`

Returns total number of tags in this group.

Return value
The number of tags.

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getCount();
```

getTags

`object getTags()`

Returns the list of all tags in group.

Return value
An array of all tags in the group.

```
var group = new Group();
project.getGroup("enginesettings", group);
var tagList = group.getTags();
for(var i = 0; i < tagList.length; i++){
      var tagName = tagList[i];
      //do something…
}
```

34.3.4 Project

This object defines the project widget. The `project` widget is used to retrieve data about the project such as tags, alarms, recipes, schedules, tags and so on. There is only one widget for the project and it can be referenced through the `project` variable.

Follow the list of properties of Project Object:

startPage

`string startPage`

The page shown when the application is started

```
var startPage = project.startPage;
project.startPage = "Page2.jmx";
```

Follow the list of methods of Project Object:

nextPage

`void nextPage()`

The script executes the next page macro.

```
project.nextPage();
```

prevPage

`void prevPage()`

The script executes the Previous page macro.

```
project.prevPage();
```

homepage

`void homePage()`

The script executes the Home page macro.

```
project.homePage();
```

loadPage

`void loadPage(pageName)`

The script executes to load the set page defined in the script.

```
project.loadPage("Page5.jmx");
```

showDialog

`void showDialog(pageName)`

The script executes to show the dialog page.

```
project.showDialog("Dialog.jmx");
```

closeDialog

`void closeDialog()`

The script executes to close the currently-opened dialog page.

```
project.closeDialog();
```

showMessage

`void showMessage( message )`

The script executes to display the message popup.

```
project.showMessage("Hi This is test message");
```

getGroup

`number getGroup( groupName, groupInstance, [callback] )`

Fast read method; this gets the values of all tags in a group.

Parameters

`groupName`	A string containing the name of the group.
`groupInstance`	The group element to be filled.
`callback`	A string containing the name of the function to be called when the group is ready.

Return value
A number value that is the status: 1 for success, 0 for fail.

```
var group = new Group();
var status = project.getGroup ("enginesettings", group);
if (status == 1) {
      var value = group.getTag("Tag1");
      if (value!=undefined) {
            // do something with the value
      }
}
```

```
var g = new Group();
var status = project.getGroup ("enginesettings", g, "fnGroupReady");
function fnGroupReady(groupName, group) {
      var val = group.getTag("Tag1");
      if (val!=undefined) {
            // do something with the value
      }
}
```

getTag

`object getTag( tagName, state, index, forceRefresh)`

`void getTag( tagName, state, index, callback, forceRefresh)`

It returns the tag value or the complete array if `index` value is -1 of the given `tagName`.

Parameters

`tagName`	A string of the tag name.
`state`	The state element to be filled.
`index`	An index if the tag is array type. -1 returns the complete array. Default is 0.
`callback`	Function name if an asynchronous read is required. Default = “”.
`forceRefresh`	Optional parameter (false as default) that force runtime to read tag value directly from device (and not from cache).

Return value
Tags value is returned. If tag is array type and index = -1 then the complete array is returned.

Remarks
For non-array tags provide index as 0.

Follow some additional details related to the use of *getTag* function with *forceRefresh* parameter. If:
- the tag to read with getTag is not in page (so not subscribed by the runtime) >> the tag is read from device irrespective of *forceRefresh* flag.
- the tag to read with getTag is present in page (so tag subscribed) & *forceRefresh* == *True* >> the tag is read from device
- the tag to read with getTag is present in page (so tag subscribed) & *forceRefresh* == *False* >> the tag is read from the tags cache at UI side.

forceRefresh parameter is working in both sync and async (with callback) mode.

NOTE *The value read from getTag when forceRefresh parameter is not set could be obsolete.*

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
//
//for non array type
//tags index is not considered, so can be left as 0
//
if (value!=undefined) {
      //...do something with s
}
```

```
var state = new State();
project.getTag("Tag1", state, -1, "fnTagReady");

function fnTagReady(tagName, tagState) {
      if (tagName=="Tag1") {
            var myValue = tagState.getValue();
      }
}
```

setTag

`number setTag( tagName, tagValue, [index], [forceWrite] )`

Sets the given Tag in the project. Name and value are in a string.

Parameters

`tagName`	A string of the tag name.
`tagValue`	An object containing the value to write.
`index`	An index if tag is array type. Set -1 to pass complete array. Default is 0.
`forceWrite`	A boolean value for enabling force write of tags, the function will wait for the value to be written before it returns back. Default is **`false`**.

Return value

Interger value for denoting success and failure of action when forceWrite is **`true`**. A **0** means success and **-1** means failure. If forceWrite is **`false`**, returned value will be **`undefined`**.

```
var val = [1,2,3,4,5];
var status = project.setTag("Tag1", val, -1, true);
if (status == 0) {
      // Success
} else {
      // Failure
}
```

```
var val = "value";
project.setTag("Tag1", val);
```

getRecipeItem

`object getRecipeItem (recipeName, recipeSet, recipeElement)`

Gets the value of the given recipe set element.

Parameters

`recipeName`	A string representing the recipe name.
`recipeSet`	A string representing the recipe set, can be either the recipe set name or 0 based set index.
`recipeElement`	A string representing the recipe Element, can be either the element name or 0 based element index.

Return value
An object with the value of the recipe. `undefined` is returned if invalid. If of type array, an array object type is returned.

```
var value = project.getRecipeItem("recipeName", "Set", "Element");
```

setRecipeItem

`number setRecipeItem (recipeName, recipeSet, recipeElement, value )`

Gets the value of the given recipe set element.

Parameters

`recipeName`	A string representing the recipe name.
`recipeSet`	A string representing the recipe set, can be either the recipe set name or 0 based set index.
`recipeElement`	A string representing the recipe Element, can be either the element name or 0 based element index.
`value`	An object containing the value to store in the recipe. It can be an array type too.

Return value
Interger value for denoting success and failure of action. A ‘0’ means success and ‘-1’ means failure.

```
var val = [2,3,4];
project.setRecipeItem("recipeName", "Set", "Element", val);
if (status == 0) {
      // Success
} else {
      // Failure
}
```

downloadRecipe

`void downloadRecipe (recipeName, recipeSet )`

Downloads the recipe set to corresponding tag.

Parameters

`recipeName`	A string representing the recipe name.
`recipeSet`	A string representing the recipe set, can be either the recipe set name or 0 based set index.

```
project.downloadRecipe("recipeName", "Set");
```

uploadRecipe

`void uploadRecipe (recipeName, recipeSet )`

Uploads the value of tags into the provided recipe set.

Parameters

`recipeName`	A string representing the recipe name.
`recipeSet`	A string representing the recipe set, can be either the recipe set name or 0 based set index.

```
project.uploadRecipe("recipeName", "Set");
```

launchApp

`void launchApp( appName, appPath, arguments, singleInstance)`

Execute an external application.

Parameters

`appName`	A string contains the application name
`appPath`	A string contains the application path, it must be an absolute path.
`Arguments`	A string contains the arguments to send to application executed.
`singleInstance`	true=single instance allowed, false allow multiple instance

```
project.launchApp("PDF.exe","\\Flash\\QTHMI\\PDF","\\USBMemory\\file.pdf","true"
);
```

printGfxReport

`void printGfxReport( reportName, silentMode)`

Prints the graphic report specified by reportName.

Parameters

`reportName`	A string containing the report name
`silentMode`	true = silent mode (avoids to show printer settings dialog)

```
project.printGfxReport("Report Graphics 1", true);
```

printText

`void printText( text, silentMode)`

Print a fixed text.

Parameters

`text`	A string to print
`silentMode`	true = silent mode (avoids to show printer settings dialog)

```
project.printText("Hello I Am Text Printing",true);
```

emptyPrintQueue

`void emptyPrintQueue()`

Empties the print queue. Current job will not be aborted.

```
project.emptyPrintQueue();
```

pausePrinting

`void pausePrinting();`

Suspends printing operations. Will not suspend the print of a page already sent to the printer.

```
project.pausePrinting();
```

resumePrinting

`void resumePrinting();`

Resumes previously suspended printing.

```
project.resumePrinting();
```

abortPrinting

`void abortPrinting();`

Aborts current print operation and proceed with the next one in queue. This command will not abort the print of a page already sent to the printer.

```
project.abortPrinting();
```

printStatus

`project.printStatus;`

Returns a string representing current printing status:

- *error*: an error occurred during printing
- *printing*: ongoing printing
- *idle*: system is ready to accept new jobs
- *paused*: printing has be suspended

```
var status = project.printStatus;
project.setTag("PrintStatus",status);
```

printGfxJobQueueSize

`project.printGfxJobQueueSize;`

Returns the number of graphic reports in queue for printing.

```
var gfxqueuesize = project.printGfxJobQueueSize;
project.setTag("printGfxJobQueueSize",gfxqueuesize);
```

printTextJobQueueSize

`project.printTextJobQueueSize;`

Returns the number of text reports in queue for printing.

```
var textjobqueuesize = project.printTextJobQueueSize;
project.setTag("printTextJobQueueSize",textjobqueuesize);
```

printCurrentJob

`project.printCurrentJob;`

Returns a string representing current job being printed

```
var currentjob = project.printCurrentJob;
project.setTag("printCurrentJob",currentjob);
```

printActualRAMUsage

`project.printActualRAMUsage;`

Returns an estimate of RAM usage for printing queues

```
var myVar = project.printActualRAMUsage;
alert(" actual ram usage is   "+ myVar);
```

printRAMQuota

`project.printRAMQuota;`

Returns the maximum allowed RAM usage for printing queues

```
var ramquota = project.printRAMQuota;
project.setTag("printRAMQuota",ramquota);
```

printActualDiskUsage

`project.printActualDiskUsage;`

Returns the spool folder disk usage (for PDF printouts)

```
var myVar1 = project.printActualDiskUsage;
alert(" actual disk usage is   "+ myVar1);
```

printDiskQuota

`project.printDiskQuota;`

Returns the maximum allowed size of spool folder (for PDF printouts)

```
var diskquota = project.printDiskQuota;
project.setTag("printDiskQuota",diskquota);
```

printSpoolFolder

`project.printSpoolFolder;`

Returns current spool folder path (for PDF printouts)

```
var spoolfolder = project.printSpoolFolder;
project.setTag("printSpoolFolder",spoolfolder);
```

printPercentage

`project.printPercentage;`

Returns current job completion percentage (meaningful only for multipage graphic reports)

```
var percentage = project.printPercentage;
project.setTag("printPercentage",percentage);
```

34.3.5 State

Class for holding state of a variable acquired from the controlled environment. Beside value itself, it contains the timestamp indicating when the value is collected together with flags marking quality of the value.

Follow the list of methods for State object:

getQualityBits

`number getQualityBits()`

Returns an integer - a combination of bits indicating tag value quality.

Return value
A number containing the quality bits.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
var qbits = state.getQualityBits();
```

getTimestamp

`number getTimestamp()`

Returns time the value was sampled.

Return value
A number containing the timestamp (for example 1315570524492).

Remarks
Date is a native JavaScript data type.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
var ts = state.getTimestamp();
```

isQualityGood

`boolean isQualityGood()`

It returns whether value contained within this State object is reliable.

Return value
A Boolean `true` if quality is good, `false` otherwise.

```
var state = new State();
```

```
var value = project.getTag("Tag1", state, 0);
if (state.isQualityGood()) {
      // do something…
}
```

34.4 Keywords

Global objects are predefined and always available objects that can be referenced by the names listed below.

page

`object page`

It references the page object for the current page.

```
function btnStd04_onMouseRelease(me) {
      var wgt = page.getWidget("rect1");
      var name = wgt.objectName;
}
```

project

`object project`

It references the project Widget.

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getCount("Tag1");
```

34.5 Global Functions

print

`void print( message )`

It prints a message to the HMI Logger window.

Parameters

`message` A string containing the message to display.

```
print("Test message");
```

alert

`void alert( message )`

It displays a popup dialog with the given message. The user must press the OK button in the dialog to continue with the execution of the script.

Parameters

`message` A string containing the message to display.

NOTE *The alert function is often used for debugging JavaScript routines.*

```
alert("Test message");
```

34.6 Limitations

Widgets cannot be instantiated from JavaScript. The Widgets can only be accessed and changed. If you need additional Widgets on the page, you can add hidden Widgets on the page, and show or position them from JavaScript.

34.7 Debugging of JavaScript

JMobile Studio and Runtime include a JavaScript debugger to allow user to debug problems.

There're two types of debuggers:

- **Runtime debugger**: a debugger running directly into target device (HMI panel)
- **Remote debugger**: a debugger running on a remote PC connected to target device via Ethernet (usually PC with JMobile Studio)

To enable the debugging mode, in the **Advanced Properties** of a **Page** set **JavaScript Debug** to **True** as shown in the below figure.

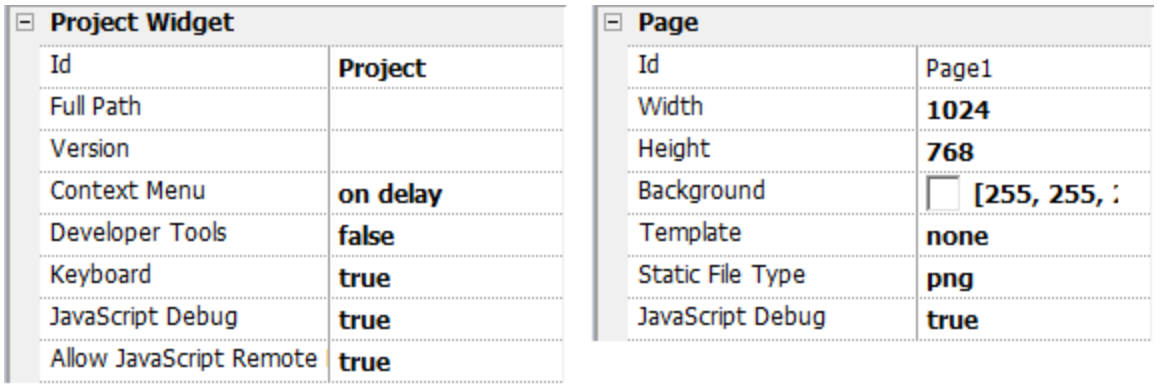


Figure 294

For schedulers and alarms debugging, enable **JavaScript Debug** in **Project properties**.

In Runtime, when the events are called, the script debugger will show the debug information (as shown in the figure below). In the box **Locals** you can inspect all available variables and elements.

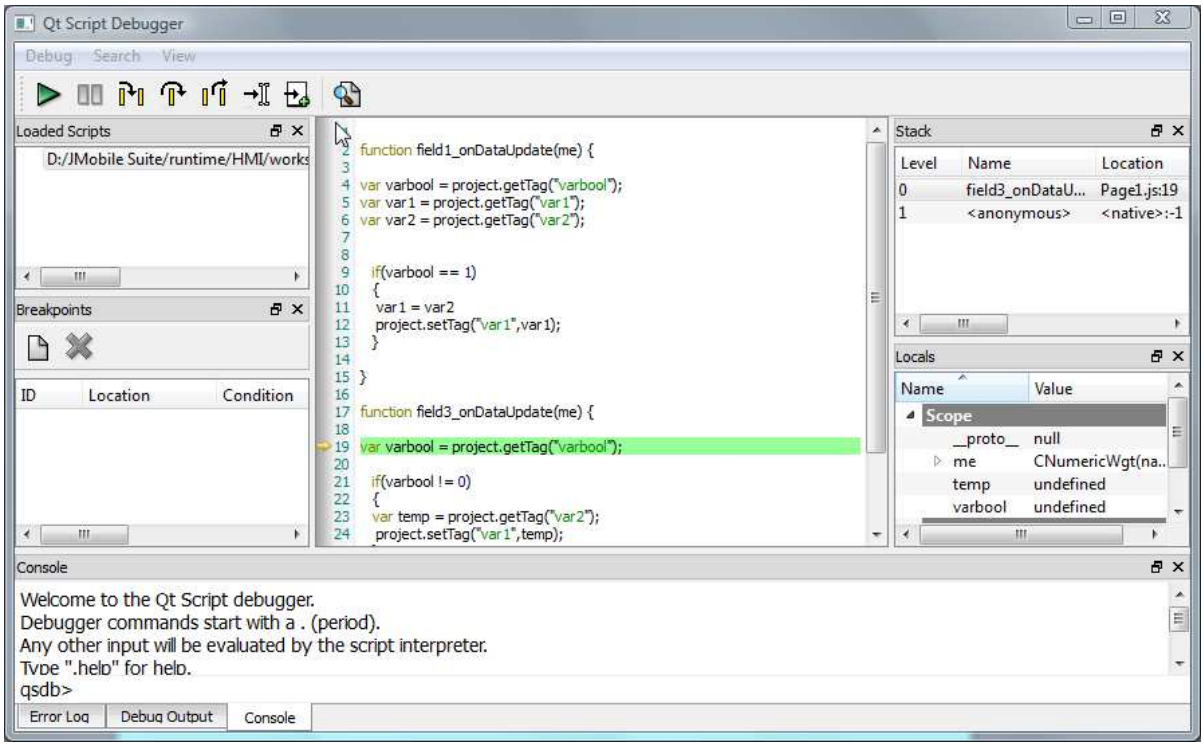


Figure 295

For a complete reference guide about Qt Script Debugger you can open the following link in your browser:

NOTE *For UN20 target (WCE MIPS hmi panels), local debugger has been disabled. However, remote debugger is available to debug JS from a PC connected to HMI panel via ethernet.*

NOTE Remote debugger not supported in JMobile Client and ActiveX.

http://qt-project.org/doc/qt-4.8/qtscriptdebugger-manual.html

34.7.1 Remote JavaScript Debugger

Remote JS debugger can be opened directly from JMobile Studio **Run** -> **Start JS Remote Debugger** or from icon in toolbar.

To start remote debugging, proceed as follow:

1. Download project with **Allow JavaScript Remote** enabled in project properties and **JavaScript Debug** enable in all pages where debugging is required.
2. Once started, runtime shows waiting for remote debugger as shown below:

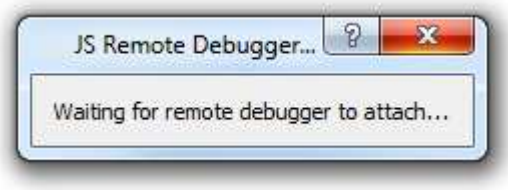
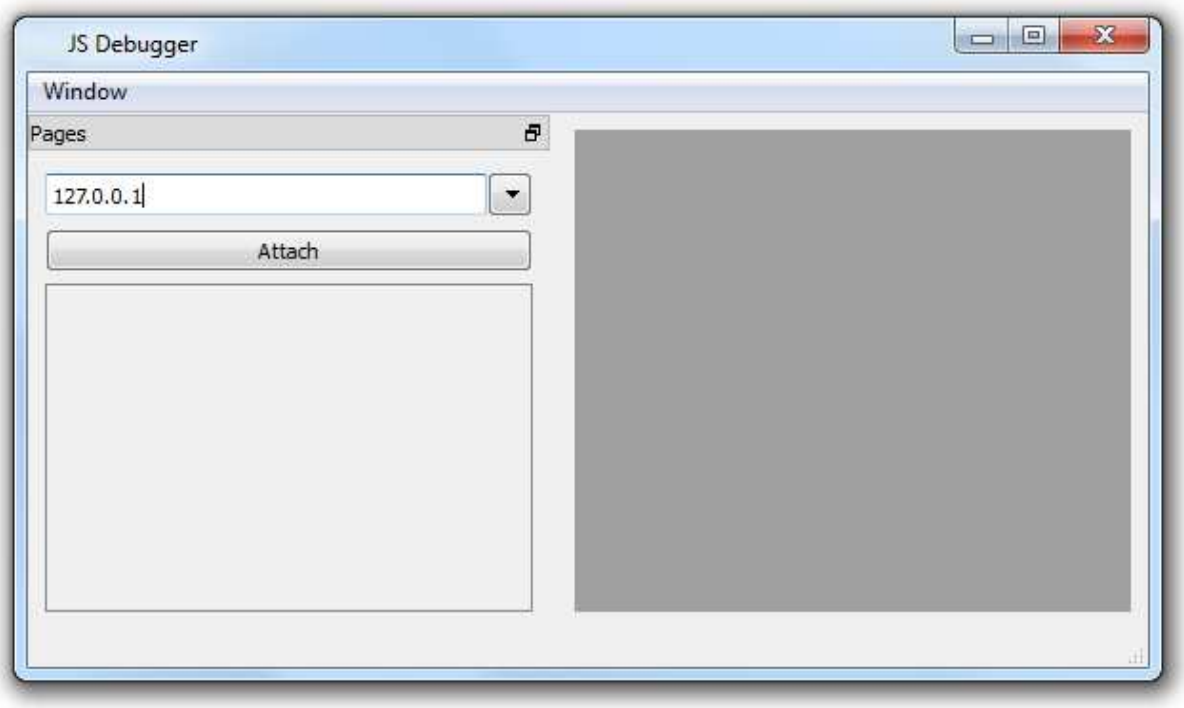


Figure 296

3. In JS Debugger window, select IP of the target and click **Attach** to connect debugger to the target.



Remote JS debugger require port 5100/TCP in the runtime side.

NOTE *For UN20 target (WCE MIPS hmi panels), local debugger has been disabled. However, remote debugger is available to debug JS from a PC connected to HMI panel via Ethernet.*

NOTE Remote debugger not supported in JMobile Client and ActiveX.

Figure 297

35 JM4Web HTML5-based Access

JM4Web allows users to access HMI projects from a remote web browser running on a computer or on a mobile device like a tablet or a phone. With JM4Web, users can create a web project tot show at a remote location the same graphical display shown on the HMI panel. JM4Web projects are based on HTML5 technology which means that no plug-ins or external software is needed for displaying the information.

This document assumes that you have a basic understanding of how to operate the web browser on your mobile devices as well as how to set up a connection to the HMI panel where the server is running. For example, you may need to set-up Wi-Fi access if you are working with tablet or phone devices to access the JM4Web pages on the HMI panel.

35.1 Supported Browsers/Platforms

35.1.1 PC

JM4Web works with all modern web browsers. The following browsers have been tested for compatibility with JM4Web:

- Mozilla Firefox 3.6+
- Microsoft Internet Explorer 9+
- Apple Safari 5+
- Google Chrome 6+

Figure 298

35.1.2 Tablets / Phones

JM4Web works with most tablet and phone devices. The following tablets have been tested for compatibility with JM4Web:

- iOS 4.1+ - Mobile Safari
- Android 2.1+ - Android Webkit

Figure 299

35.2 Generating JM4Web Pages

You can use any of your existing HMI projects or you can make a new project using JM4Web. If you are working with an existing project, any widgets and features can be used in JMobile Studio; however, not all features are currently available with the JM4Web option. If the project includes a feature that is not available, JM4Web will still work correctly but the feature will not be available on the remote client device. Following chapters will list supported features in JM4Web and existing limitations.

35.2.1 Exporting Pages

HMI pages in your project must be exported before they are available as JM4Web pages. You can export pages from the current project by clicking on the menu item **Run > Export Web Project**

NOTE *You must repeat the Export step after any change to your project. The only way changes are available in JM4Web is after the Export step and the project is downloaded to the panel.*

35.2.2 Select Pages to Export

When exporting, you have the option to select which Web pages you wish to export. By default all project and dialog pages are selected for export. You can click on the check boxes next to each page name to enable/disable the page export as shown below. You can also click on the **All/None** check box to select or de-select all pages at once.

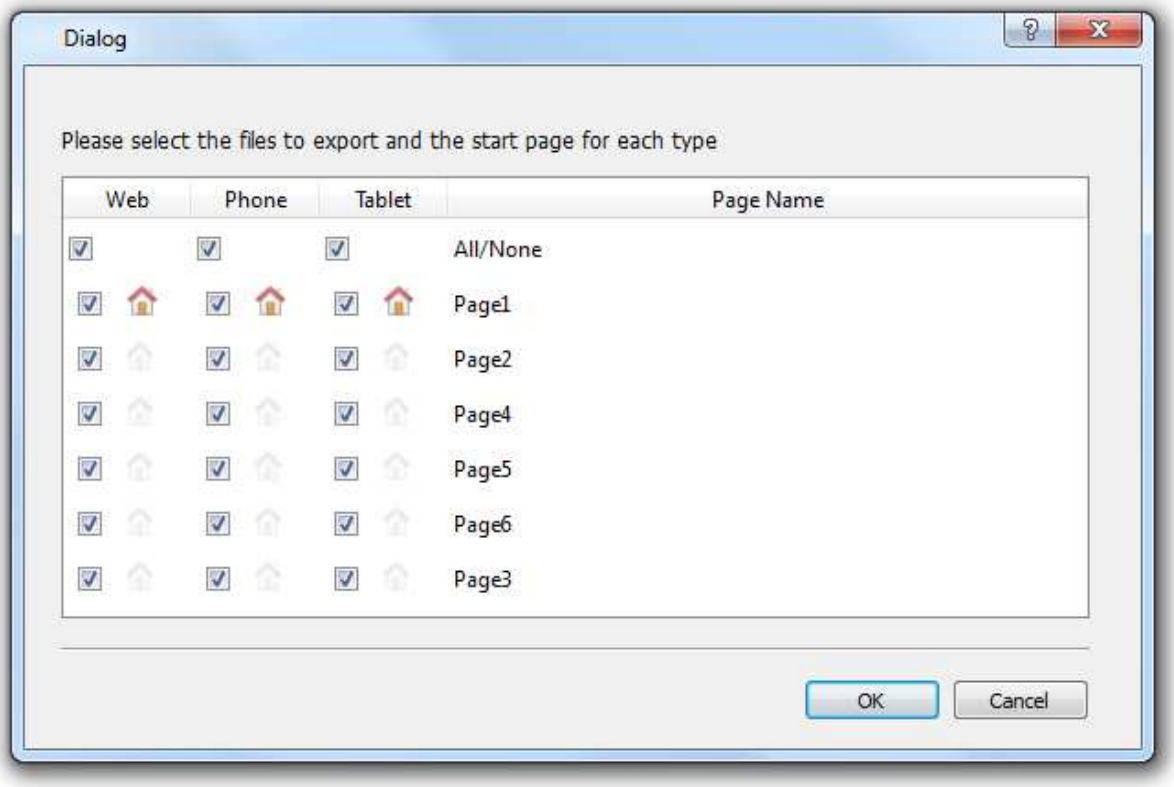


Figure 300

JM4Web supports 3 platforms: **Web**, **Phone** and **Tablet**. The **Web** platform is intended for desktop web browsers, the **Tablet** platform is intended for tablet devices and the **Phone** platform is intended for smart phone devices. By providing multiple platforms, users can create different pages for different platforms and provide different pages designed for different screen sizes. For example, you may want to create a set of smaller pages (e.g. pages the size of a mobile phone screen) in your project for phones whereas you will use full size pages for desktop web browsers and tablets.

If you have created different pages for each platform, the Export dialog allows you to select the pages that correspond to each platform. If you have a single set of pages in your project, simply click on the check boxes for each platform and the same page will be used for each platform.

35.2.3 Setting Home Page

The start page or home page of the JM4Web project defines the first page that is shown in the browser of each platform type. The home page is important because it defines the starting point for your web application. Pages that are shown after home page depend on the how the project links other pages.

For example, if you have set of pages for a phone, you can set the start page to a page that is sized appropriately for a mobile phone display. If this phone page only contains links to other phone pages, the user will only see phone pages when accessing the JM4Web project from a phone.

To select the home page for each device, click on the home icon next to the check box. When the home page is selected for a given page, you will see a full color home icon next to the page check box. All other home icons will be grayed. Only one home page can be active for each platform.

An example is shown in figure. The page **Page1** has been set as the home page for Phone and Tablet while **Page2** is the home page for desktop web browsers.

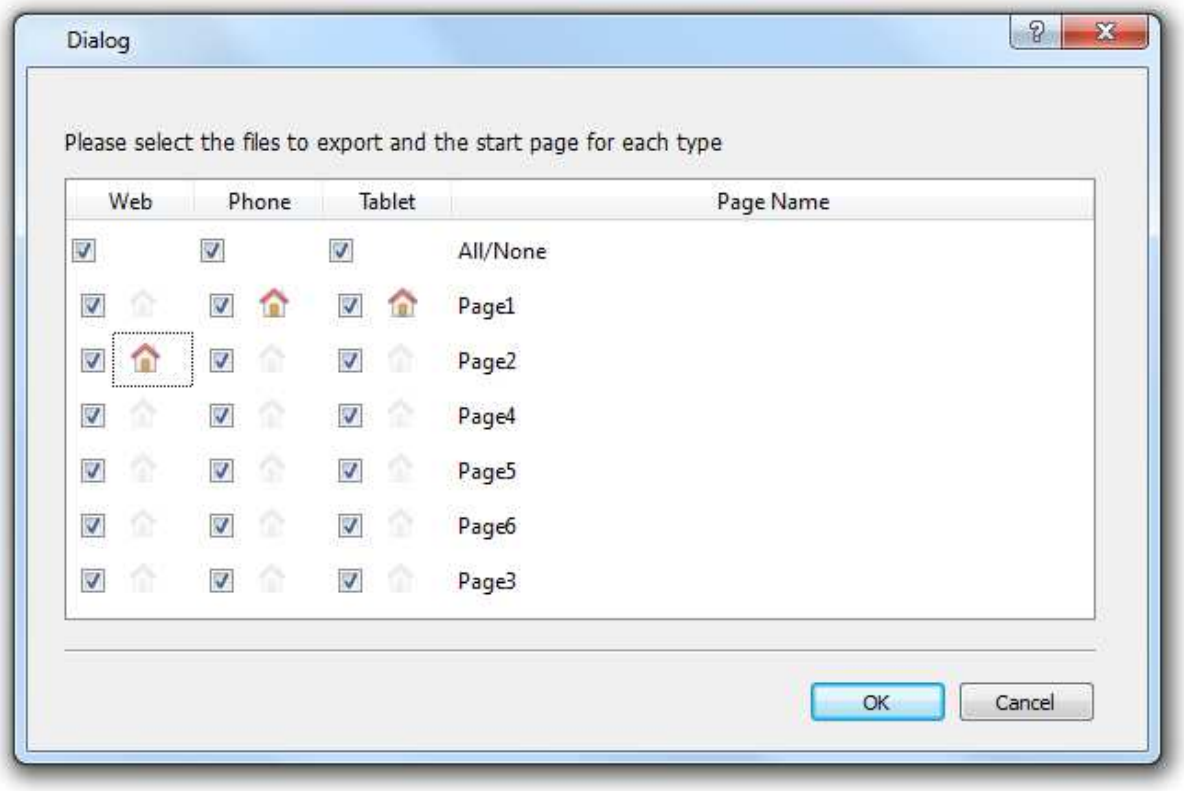


Figure 301

35.2.4 Start JM4Web Export

After selecting the pages and home page for each platform, click on **OK** button to start the JM4Web page generation.

Once the export has been started, JMobile Studio will show a progress bar that indicates the status of each page as it is being processed. If you have many pages in the project, this process can take several minutes.

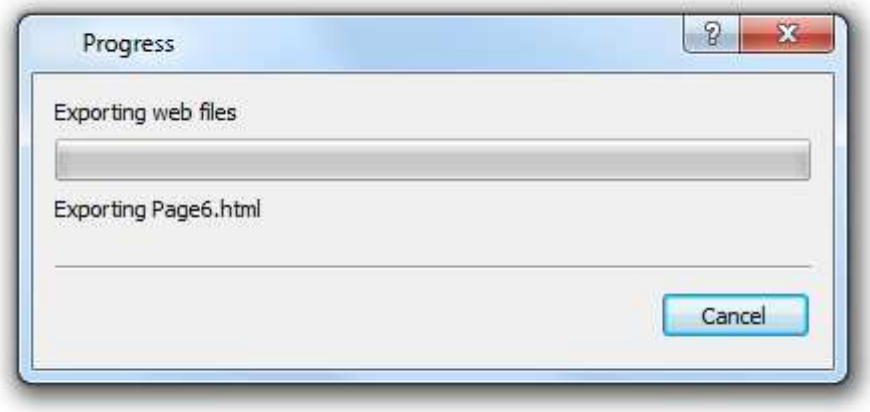


Figure 302

After the Export has been completed, the progress dialog will close automatically and the JM4Web application is ready to be downloaded to the target.

35.3 Testing the JM4Web Project

There are two methods available for testing your JM4Web project. You can use the online simulator or you can open a standalone web page directly from a browser. The following sections describe both approaches.

35.3.1 Using the Online Simulator

JMobile Studio includes an web server in the online simulator. You can start the simulator and access your JM4Web project from a web browser. The pages will be served from the simulator. Here are the steps to run a project from the simulator:

1. Create and export your project as described in previous chapter.
2. Click the **Start Simulator** button in the JMobile Studio workspace toolbar. This will start the simulator. You will see the project running in a separate window on the screen.
3. Open a web browser compatible with JM4Web requirements. You can use web browsers available in your computer such as Internet Explorer, Firefox or Chrome.
4. Type http://localhost:81 in the URL bar of the browser. This URL tells the web browser to read web pages from your local computer. The *localhost* name indicates the local computer and *:81* tells the browser to use port 81. The online simulator in JM4Web uses port 81 by default so that it does not interfere with any other web servers or applications on your computer.
5. You will see the login screen the same as you would see if you were running the web project from a panel. Enter the login credentials and you will see the first page of the project. When running from the simulator you will be executing web pages the same way that you would if you were running directly from the HMI panel.

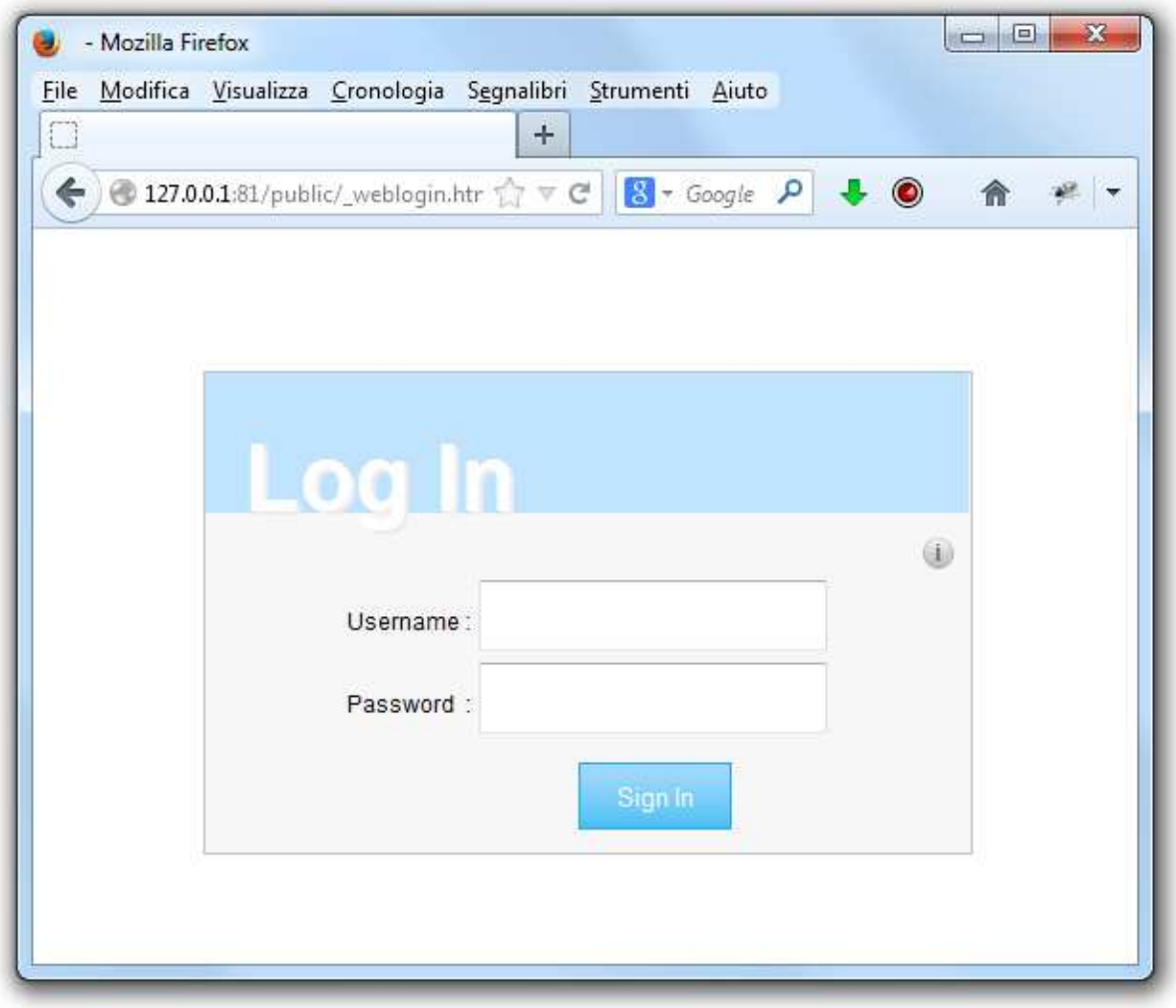


Figure 303

6. At this point, you can test the web project in the browser.
7. If you want to make changes, you can change the project pages in JMobile Studio and re-export the project. You do not need to restart the simulator at this point. You can simply click the refresh button on the browser and your changes will appear. However, if you switch from one project to another project, you need to restart the simulator for the new project.

NOTE *You need to re-export the web project after making changes to the project in order for the changes to appear in the web browser.*

35.4 Downloading the JM4Web Project

After testing the JM4Web pages, the next step is to download the project to the desired target device. The JM4Web project is downloaded at the same time the normal JMobile Studio project is downloaded. To download JM4Web to the panel, you perform the same steps as you would to download a normal JMobile Studio project. Both the HMI panel project and JM4Web web project are downloaded at the same time.

To perform the download, click on the **Run > Download To Target** menu item. You can also initiate the download process by clicking the **Download To Target…** button in the JMobile Studio toolbar.

After the download process is complete, the Download dialog box will be removed and the HMI project will be automatically started on the HMI panel. The JM4Web project is now ready to be used.

35.4.1 Running JM4Web from a Browser

After the successful download of the project to your panel and the panel is running, your JM4Web pages are ready for access through a web browser from any device of your choice.

To run the project, open the browser and type in the IP address of your HMI panel. You will see the JM4Web Login page.

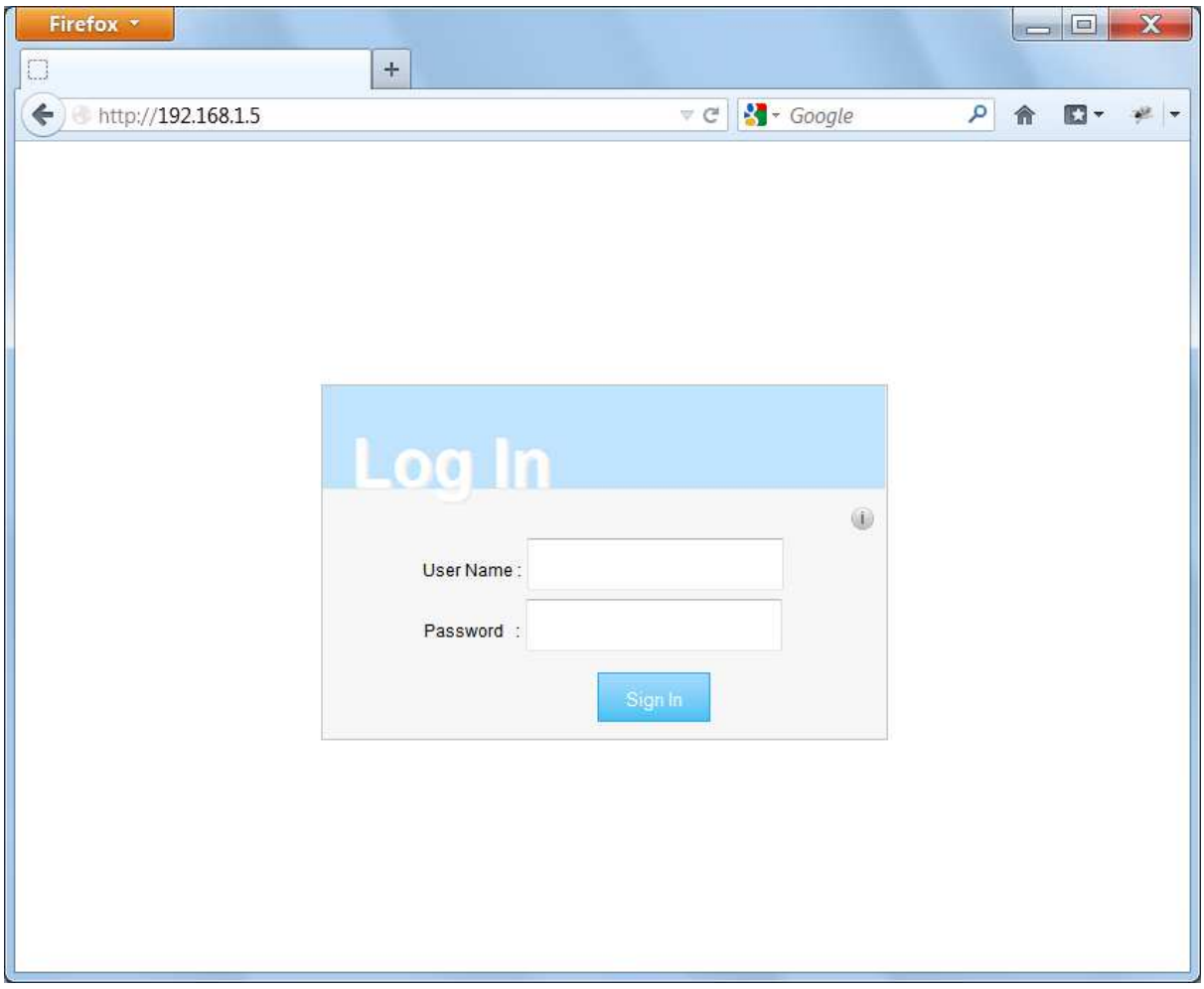


Figure 304

Enter **User Name** and **Password** and click on Sign In. The User Name and Password are the credentials that you have defined in your project using the User Management features.

If the correct credentials are entered, you will be redirected to the home page that you have selected during the page export. An example of the home page is shown in the figure below.

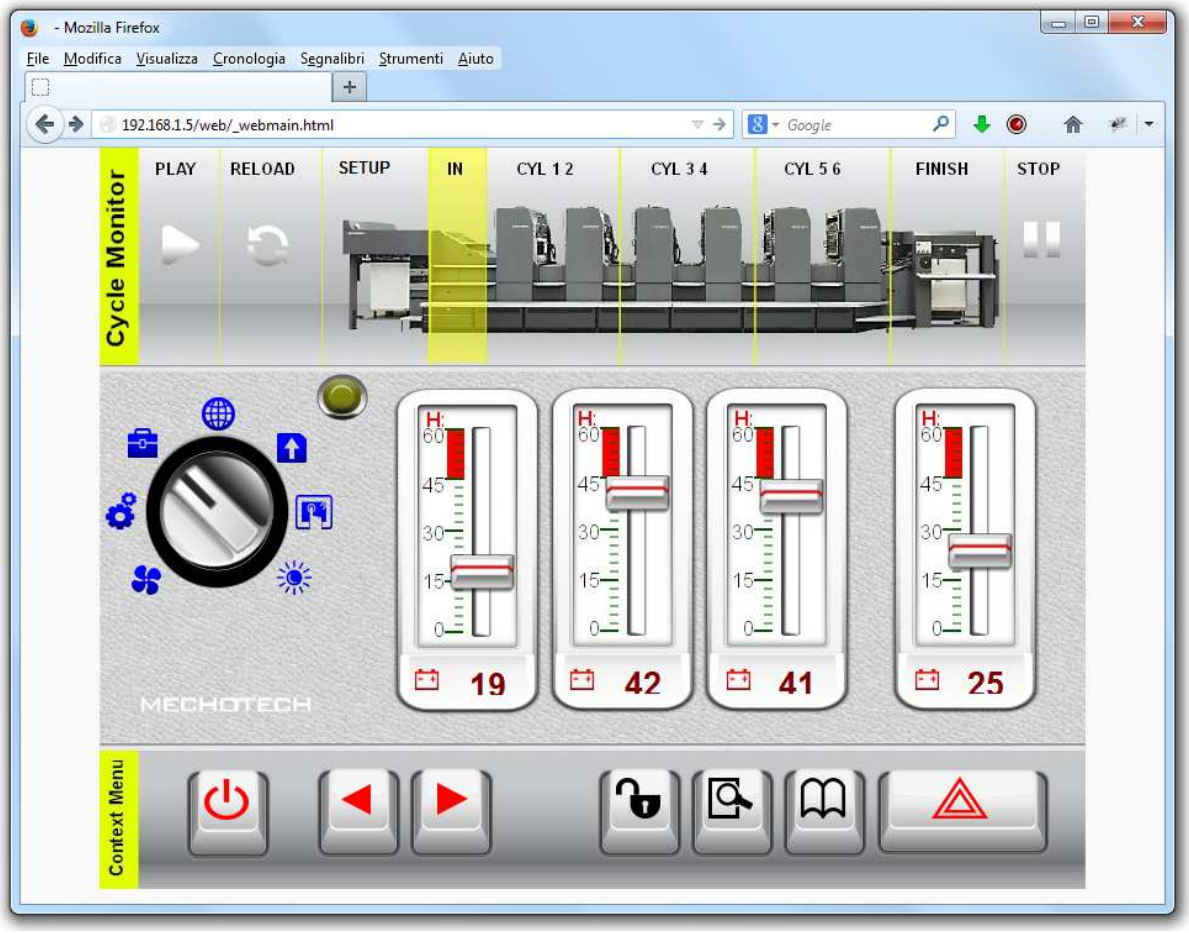


Figure 305

You can interact with the project using the browser in the same way you interact with a panel when touching the screen. You can click on buttons to change pages, view indicators and gauges, drag slider handles to change values, etc. The JM4Web project will manage all communications with the web server while you are interacting with the HMI panel remotely.

35.5 JM4Web Connectivity Notes

The following sections describe issues of JM4Web that may appear when connecting remotely to a HMI panel.

35.5.1 Server Disconnect

Since JM4Web runs remotely from the HMI panel, it is possible for the server to disconnect from the browser; for example if the server is stopped or the network cable is unplugged. If JM4Web is disconnected, a ‘disconnect’ icon will appear in a toolbar on top of the JM4Web page as shown below. Once the server is back online, the red circle-bar icon will be removed indicating normal communications with the panel.

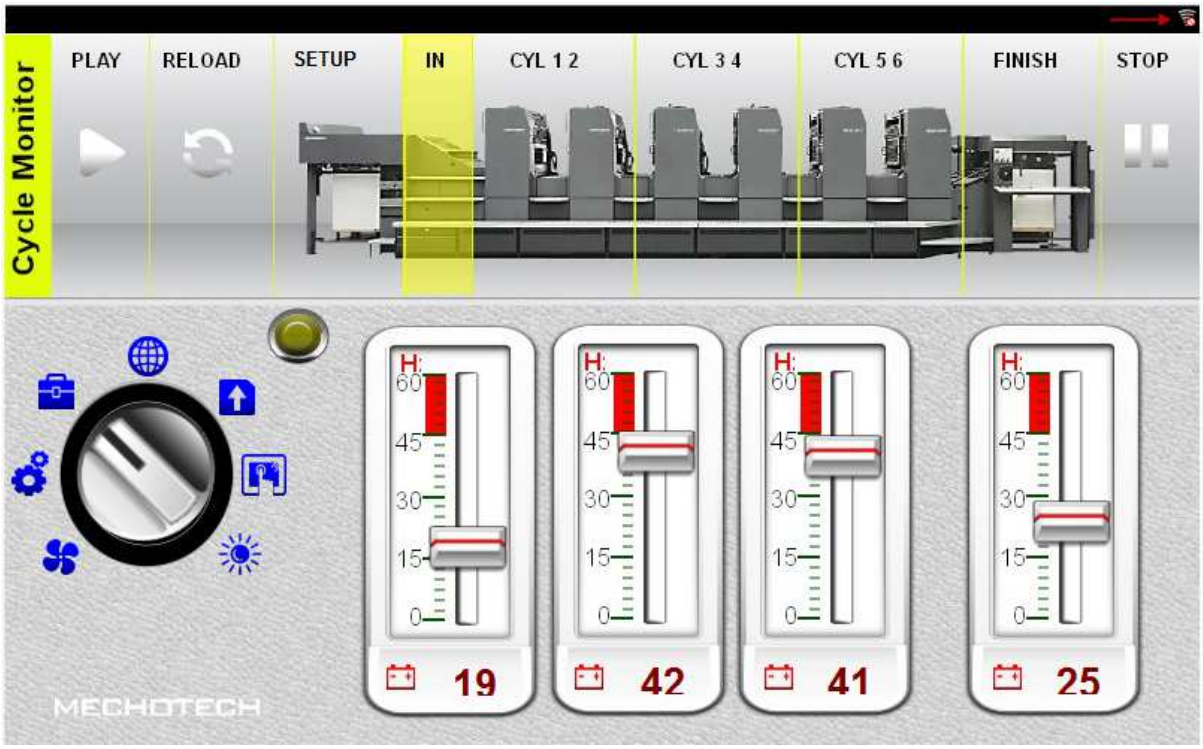


Figure 306

Note that if the server is disconnected, changes may still be made in JM4Web pages but changes will not result in any effect in the server. For example, if a button is pressed to change the tag value, the tag value may appear changed on the client, but it will not be updated in the server. Once the server connection is restored, the server based value will be updated in the JM4Web pages.

35.5.2 Inactivity Timeout

JM4Web will require you to re-enter your login credentials if the browser has been inactive for several minutes. If no activity has occurred for 10 minutes, the login screen will reappear and you need to enter your login credentials to continue operation. This feature has been provided as a safety measure so that unauthorized users cannot operate if the remote device is not being actively used.

35.5.3 User Session Termination

A user session termination/completion can come either from the server or based on the actions by the user.

At times the server may send a response to the client (browser) to perform the login process. In such a case the user will be redirected to the login page. The user will be redirected back to the last page they were in.

If the user clears the browser cache or browser cookies, the server will ask for user login. After a successful login, user will be redirected back to the project page that was active before.

NOTE *If the user is in a dialog page during a page redirection to the login page then after the successful login, the user will be redirect back to the page from which the user had opened the dialog page.*

35.5.4 Non-Active JM4Web Project

You can be in a situation in which you have a JM4Web page in your browser, which comes from a project that is no longer active in the panel.
In such a case, you will get a confirmation box through which you can redirect to the active project, if you wish.
You can choose whether you want to redirect to the active project or stay in the non-active project.
If the you want to redirect to the active project then click on **OK** button and you will be redirected to the current active project page.

NOTE *This redirection is based on the assumption that the current active project has JM4Web pages in it.*

If you want to stay in the non-active project whatever actions you perform in the browser may not complete properly as the JM4Web will not perform any server-bound communications.

35.6 JM4Web Limitations

Currently not all JMobile Studio features are supported in JM4Web. For example, JM4Web only supports a subset of widgets available in JMobile Studio. The following sections define the items that are supported by JM4Web.

35.6.1 Widgets

- *Label*
- *Numeric Fields*
- *Buttons*
- *Hotspot Buttons*
- *Horizontal / Vertical Sliders*
- *Gauge / Needle*
- *Images*
- *Multistate Images (MultistateImage MultiLayer not supported)*
- *Indicators (Lights)*
- *Date/Time*
- *Bar Graphs*
- *Segmented Bar Graphs*
- *Message widget*
- *Knobs*
- *Shapes*
- *IPCamera (single frames or streaming MJPEG for devices supporting it)*

There are some known issues that you may experience while using JM4Web.

- Widget rotation is not supported.
- Not all custom fonts appear correctly in iOS devices.
- Vertical Slider reverse mode is not supported.

- Shape widgets are generated as images. As a result, the look and feel of the shape cannot be changed by attaching tag to the UI-based properties of the widget.
- ColorPalette XForm works only with selected widgets including Buttons, Images, Bar Graphs.
- Ticks, Labels and Scale widgets are not supported. However, if you set 'static optimization' to 'static' on ticks, labels and scale widgets they may be available under certain conditions.
- Widget properties with **Attach to…** dynamic behaviour may not work for all properties supported by JMobile Studio.
- Dialog pages support only modal dialogs.
- Dialog pages do not support runtime positioning and are not movable.
- Round Bar Button widget is not supported.
- User management features of widgets are not supported.
- onDataUpdate event on Page/Template Page/Dialog Pages is not supported.

35.6.2 Actions

JM4Web currently supports the following JMobile Studio actions:

- *HomePage*
- *LoadPage*
- *NextPage*
- *PrevPage*
- *LastVisitedPage*
- *WriteTag*
- *StepTag*
- *SetBit*
- *ResetBit*
- *ToggleBit*
- *ShowDialog*
- *CloseDialog*

35.6.3 Tag Attachment

JM4Web supports Tag attachments on some selected widget properties. However, JM4Web does not support tag attachment to properties that change the range of a widget. For example a tag attached to the min or max property of a Slider or Bar Graph is not supported.

35.6.4 XForms

JM4Web supports the following XForm options:

- Scaling

- Offset
- ColorPalette
- BitIndex

The *ColorPalette* functionality works for Buttons, Image and Bar graphs widgets limited to the properties Fill and Background Color.

35.6.5 Widget Event

JM4Web supports both macro actions and JavaScript actions for widgets.

NOTE *You may find some limitations in the OnDataUpdate event in certain widgets especially with its initial level triggering due to page loading delays when running on a remote client.*

35.6.6 Page Events

JM4Web supports both macro actions and JavaScript actions for pages with the following limitations:

- Page OnDataUpdate event is not supported.
- Dialog page OnDataUpdate event is not supported.
- Template page OnDataUpdate event is not supported.

35.6.7 JavaScript API

JM4Web does not support JavaScript functions.

35.6.8 Keypads

JM4Web provides keypad support for Numeric widgets only. Custom keypads are not supported. Numeric widgets with a Read/Write or Write mode will display the keyboard of the device.

The user can click on the Numeric widget and the widget will show a textbox in which the new value can be inserted.

After inserting the value the user can either press Enter key / Enter key equivalent in touch devices or the user can click on the Save button shown below the text box to make the newly inserted value permanent. Only meaningful numbers will be accepted during the save process. Anything else will be ignored and will not result in a value change.

Figure 307

35.6.9 Dialog Page

JM4Web supports Dialog pages. You can show them and close them based on the **ShowDialog** and **CloseDialog** macro actions. However Dialog pages are not movable.

35.6.10 User Management

JM4Web uses a login mechanism which will verify user credentials on the server. The user name and password are based on the user credentials defined in User Management.

JM4Web does not support individual security settings applied to widgets or pages.

35.6.11 Maximum Concurrent User Connections

The web server in the HMI panel supports three (3) concurrent connections at a time. If you try to connect to the server from more than 3 remote browsers, only the first 3 connections will be permitted.

35.7 Troubleshooting and FAQ

35.7.1 Enable JavaScript

JM4Web requires JavaScript to provide interactivity with the server and the user. JM4Web will not work if JavaScript is disabled in your browser.

By default most browsers come with JavaScript enabled. But if you have disabled JavaScript in the past, please re-enable JavaScript before accessing JM4Web pages.

35.7.2 Browser Cache

JM4Web include resources that change infrequently such as CSS files, image files and JavaScript files. These resources take time to download over the network which increases the time it takes to load the JM4Web page in your browser. Browser caching allows these resources to be saved by a browser and used without requesting them each time to the server. This results in faster loading of JM4Web pages.

For optimal JM4Web performance, caching should be enabled in your browser. By default the browser comes with caching enabled and you do not have to do anything with it unless the cache has been disabled.

Note that JM4Web pages will work properly with disabled browser caching, however resource loading time will be slower compared with normal cached operations.

35.7.3 Using a Proxy

Some users may be accessing the JM4Web project through a proxy. The proxies may control the number of parallel connection settings for the browser. If the proxy controls the parallel connection settings then make sure that the maximum possible parallel connections (max connections) is set to a number greater than 5 and less than or equal to 10.

35.7.4 Why I'm not able to see changes in web pages?

Everytime a new web page has been included or edited into the project, we must perform a *Download to Target* action to download changes into panel.

Connecting with a web browser to the panel IP address can result in the visualization of cached pages instead of the latest downloaded pages. To avoid this behaviour you can:
- disable cache of your web browser
- force web page refresh
- by-passing browser cache

36 System Settings Tool

The System Settings tool comes with an interface based on a rotating menu, with navigation buttons at the top and bottom to scroll between the available options. The tool is shown in the figure below.

On the left side, several components and functions are highlighted and, for each of them, the right side ("Info" pane) shows the information about the current version (when applicable). In the picture below, the version of the Main OS component is shown.

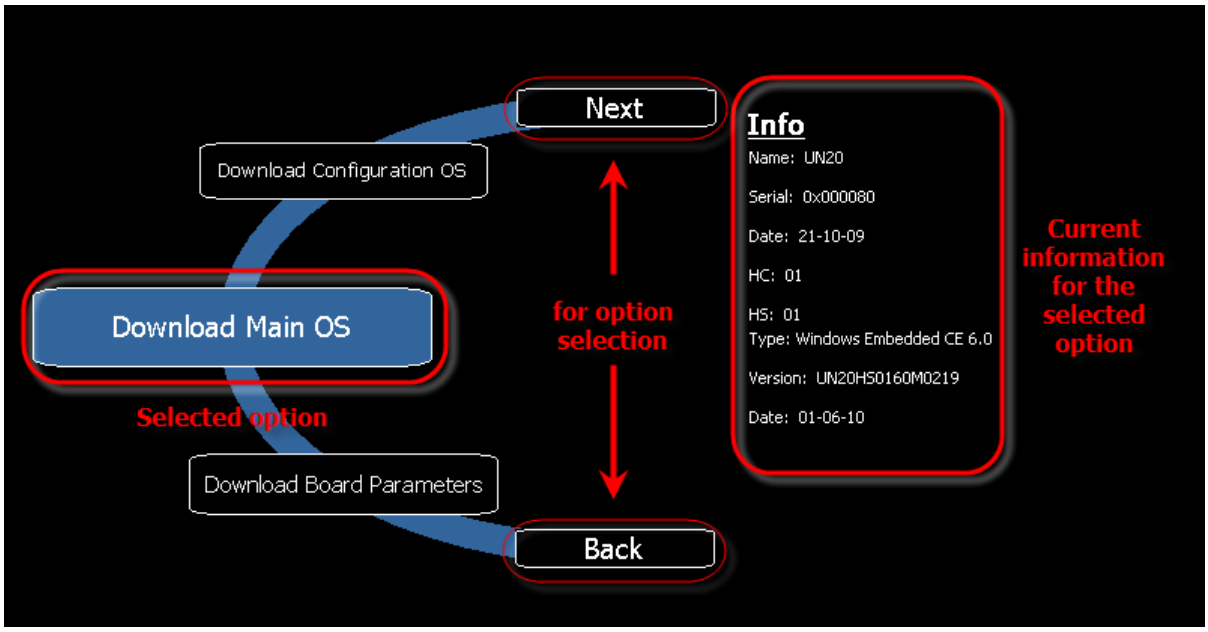


Figure 308

System Settings tool has two operating modes:

- **User Mode**
- **System Mode**.

The difference between them is the number of available options.

36.1 User Mode

User mode is a simple interface where users can get access to the basic settings of the HMI panel.

The System Settings tool is accessible at Runtime from the **context menu** by selecting the item **Show system settings**. When activated in this way, the System Settings tool always starts in **User Mode**.

The context menu can be activated by pressing and holding down a screen area without buttons or other touch sensitive elements, until the menu is displayed.

Main Items available in User Mode are:

Calibrate Touch To calibrate the touch screen if needed

Display settings	Backlight and Brightness control
Time	Internal RTC settings
BSP Settings	Operating system version, Unit operating timers: power up and activated backlight timers, Buzzer control, Battery LED control
Network	IP address settings
Plug-in List	Provides a list of the plug-in modules installed and recognized by the system; this option may not be supported by all platforms and all versions.

36.2 System Mode

System Mode is the interface of the System Settings tool with all the options enabled.

The HMI products support a special procedure for accessing the System Settings tool; the special procedure is required to start the System Settings in **System Mode**, or when the standard access procedure is not accessible for some reason.

When activated by this special procedure, the System Settings tool always starts in System Mode.

The special access to the System Settings tool can be activated with a **tap-tap sequence** over the touch screen during the power-up phase. Tap-tap consists of a high frequency sequence of touch activations, done by the simple means of finger tapping the touch screen, performed during the power up and started immediately after the device has been powered.

In addition to the options available in User Mode, the following important features are available:

Format Flash	To format the internal panel flash disk. All projects and the runtime will be erased, returning the panel to a factory new condition
Restore Factory Settings	Restore Factory Settings is used as alternative to **Format Flash** (that's a slow operation) to restore device factory settings. Options available are: **Uninstall HMI**: removes the HMI Runtime (entire qthmi folder) from the unit (if present); at the next start the panel will behave as a brand new unit. This command does not reset settings like IP, brightness or RTC **Clear System Settings**: allows you to reset the system parameters (registry settings). Files deleted are: \\Flash\\Documents and Settings\\system.hv \\Flash\\Documents and Settings\\default\\user.hv \\Flash\\Documents and Settings\\default.mky \\Flash\\Documents and Settings\\default.vol Also System Mode password is reset. **Clear internal Ctrl App**: clear current folders used by CODESYS V2.3 and CODESYS V3 internal controllers for applications • \Flash\QtHmi\RTS\APP*.* • \Flash\QtHmi\RTS\VISU*.* • \Flash\QtHmi\codesys*

- \Flash\$SysData$\codesys*

Clear sysdata settings: clear \Flash\$SysData$ folder (used by tech. supp only for problems related to display settings)

NOTE *Not all targets and BSPs contain all these options.*

Resize Image Area	Resizes the Flash portion reserved to store the splash screen image that is displayed at power up. Default settings are normally ok for all units.
Download Configuration OS	checks the actual version and upgrades the back-up operating system (see relevant chapter, for additional details)
Download Main OS	checks actual version and upgrades the main operating system (see relevant chapter for additional details)
Download Splash Image	Loads a new file for the splash screen image displayed by the unit at power up; the image must be supplied in a specific format. We suggest that you update the splash screen image directly from the JMobile Studio programming software.
Download Bootloader	Checks the actual version of the system boot loader and upgrades the system boot loader.
Download Main FPGA	Checks the actual version and upgrades the main FPGA file; this command may not be available in all platforms and versions.
Download Safe FPGA	Checks the actual version and upgrades the back-up (safe) copy of the FPGA file; this command may not be available in all platforms and versions.
Download System Supervisor	Checks the actual version and upgrades the system supervisor firmware (used for the RTC and power supply handling).

IMPORTANT *Operation with the System Settings Tool is critical and, when not performed correctly, may cause product damages requiring service of the product. Ask Technical Support for further details.*

When executed in “System Mode” the System settings also provide the “BSP Settings”. Only when recalled from the System Mode, the BSP settings show an additional tab called “**Password**” as shown in the figure below.

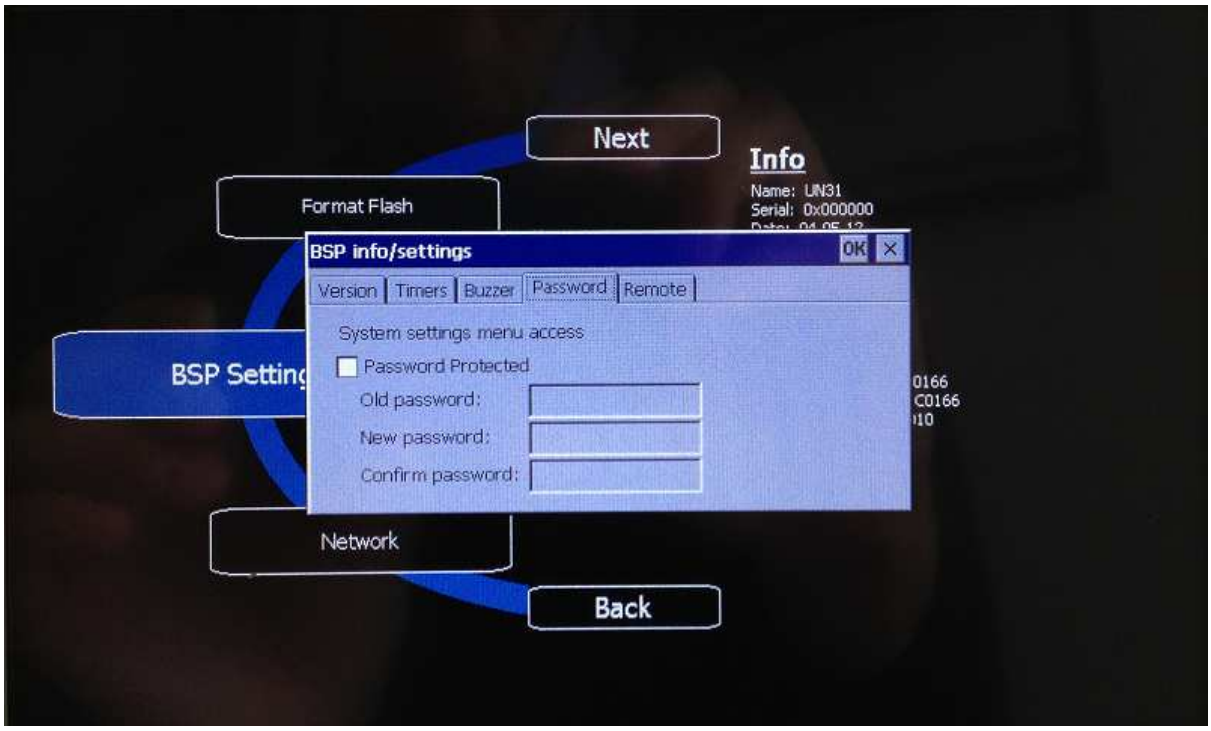


Figure 309

This function allows you to protect access to the System Settings in System mode with a password, so all the advanced and critical functions are not easily accessible to anyone.
To activate the protection, simply mark the check box “Password Protected” and specify the desired password as shown in the figure. The password must be at least 5 characters long.
If you are changing a password previously defined or disabling the protection, you are asked to provide the old password first.

NOTE *Please keep a note of the configured password in a safe place. There is no way to reset the password protection and, in case it is lost, the unit must be returned to the factory for proper reconditioning.*

When the System Settings menu is protected by a password, for each critical function you try to execute that may compromise the proper system operation, the HMI will prompt you to enter the password. If correct, the operation will proceed; if wrong, the operation will be aborted.

37 License Activation of HMI Panel Software Modules

JMobile Studio provides the interface for activating software modules in the HMI Panel. You can access the UI from **Run -> Manage Target - >License**. The interface provides two operations - **Activate Panel** and **Save License** - which needs a working Internet connection or – **Import License** – which can be done offline.

37.1 Activate Panel

Activate panel operation is used to activate the license on its corresponding panel. The panel must be connected to the network.

IMPORTANT *You must install the Runtime in the HMI Panel to activate the licenses.*

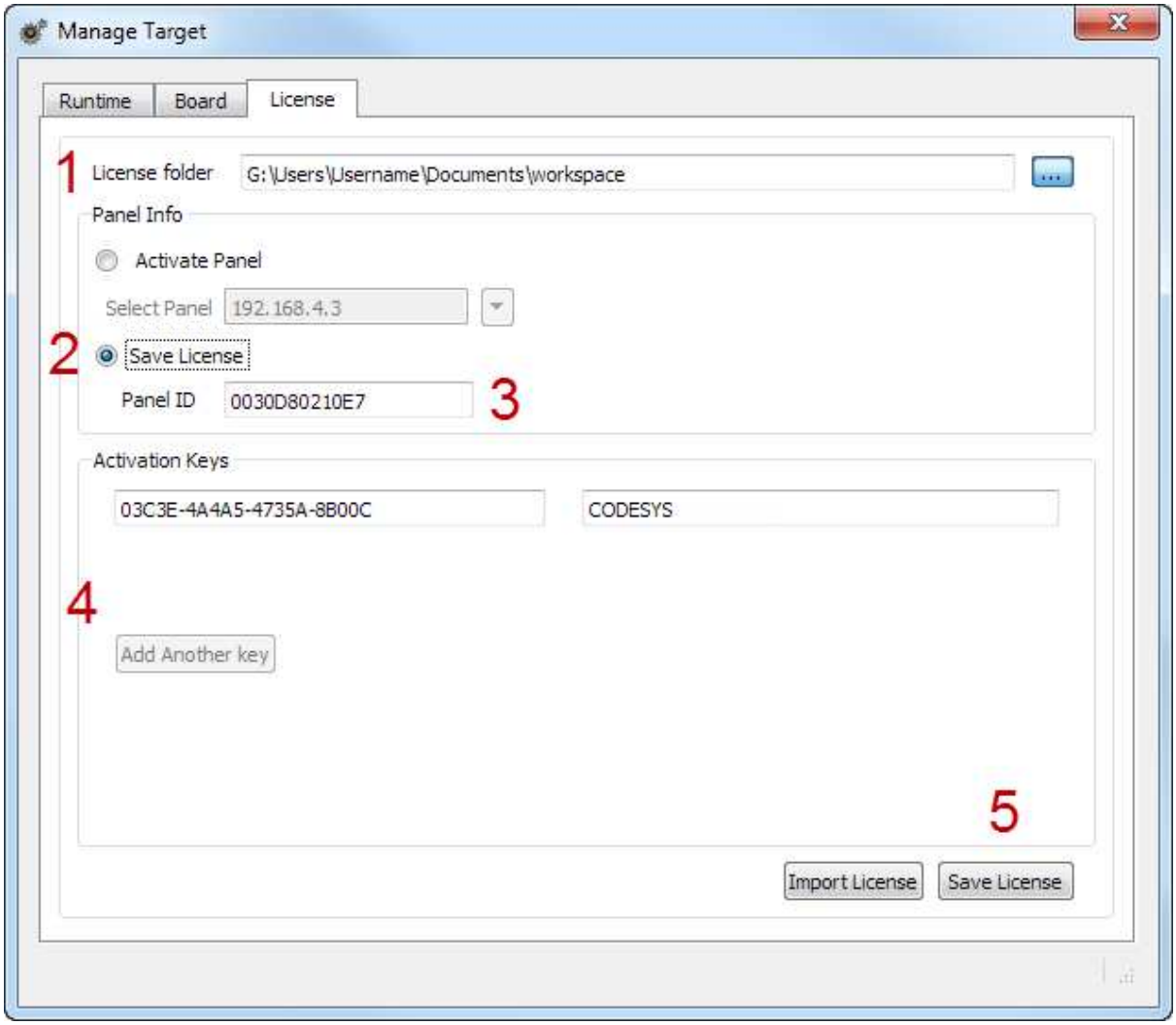


Figure 310

1. Select the location used to backup license files. From this point on, the path selected will be called **LICENSEFOLDER** in the document.
2. Select the **Activate Panel** option using the radio button.
3. Select Panel.
4. Press the **Select IP** button and it will show all of the panels connected to the network. Then select the panel on which the license is to be activated.
   Once you select the IP,
   - The Panel ID number will be shown in the **Panel ID** box.
   - If you have the backup files of previously activated or created license files in the **LICENSEFOLDER**, the activation key and enabled features will be displayed in the **Activation keys** section.
   - One empty edit box in the **Activation keys** section will be enabled for you to enter a new activation key. The **Activate Panel** button will be enabled.
5. Enter activation key.
   If you have a new activation key to be activated enter it in the empty edit box in the Activation keys section. Once you enter a new activation key The **Add Another Key** button will be enabled, so that you can add a new edit box and enter another activation key.
6. Activate panel
   Once you have completed entering the activation key, you can press the **Activate Panel** button to activate licenses on the HMI Panel.

Once you press the **Activate Panel** button
- It verifies the activation key.
- It creates a license file and stores it in the folder LICENSEFOLDER\HMI Licenses\PANELID.
- It downloads the license file to the HMI Panel and activates it.

Once the activation is completed, the enabled features or status of the activation keys will be displayed in the Feature/status list box that can be seen on the right side of the Activation keys boxes.

NOTE *After the activation is complete, to enable the license in the panel you must restart it.*

37.2 Save License

Save License operation is done to create the license file now for activation later.

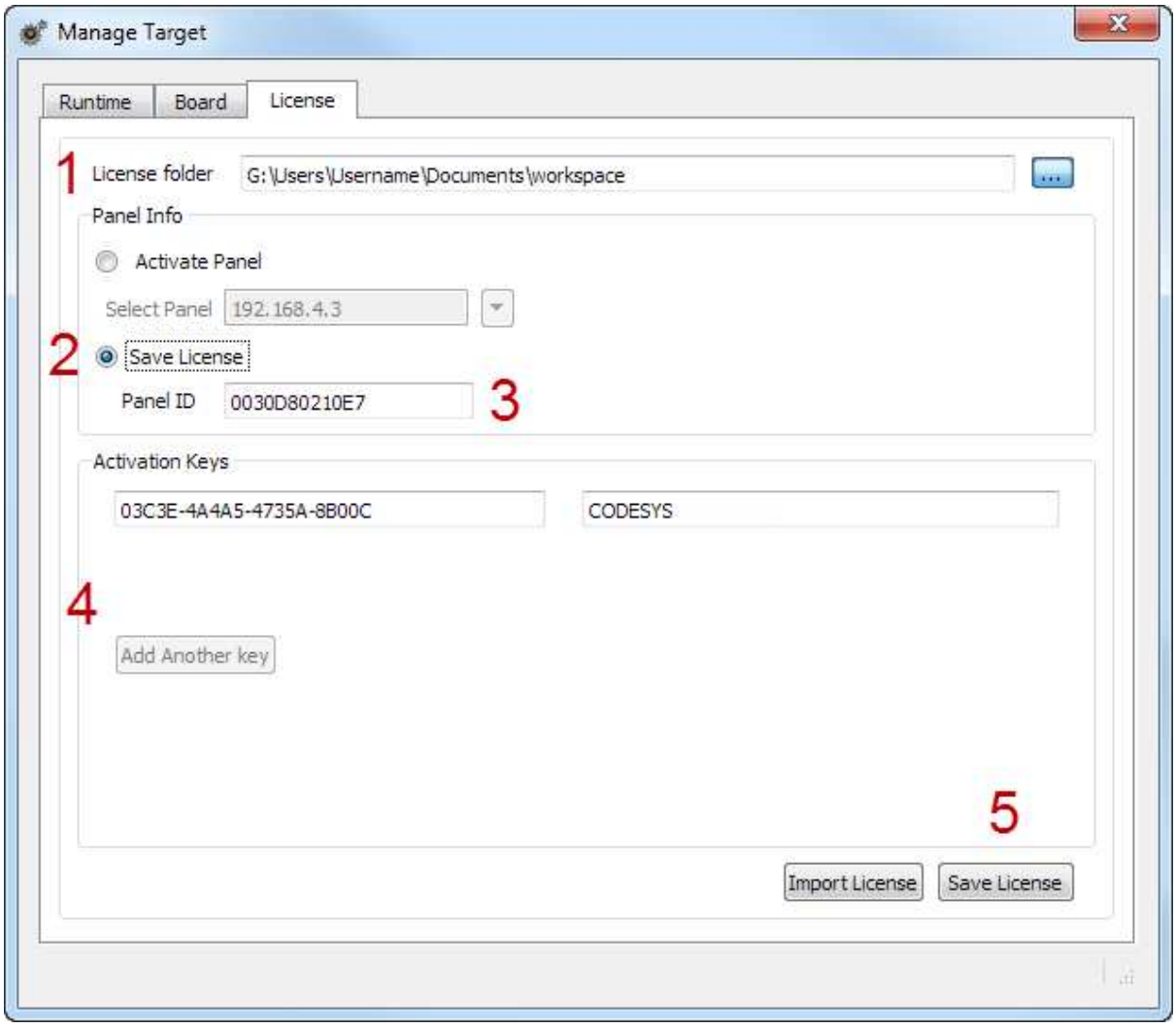


Figure 311

1. Select the location used to backup license files. From this point on the path selected will be called **LICENSEFOLDER** in the document.
2. Select **Save License** option using the radio button.
3. Enter Panel ID.
   Once you enter the Panel ID
   - If you have the backup files of previously activated or created license files in the **LICENSEFOLDER**, the activation key and enabled features will be displayed in the **Activation keys** section.
   - One empty edit box in the **Activation keys** section will be enabled to enter a new activation key.
   - The **Save License** button will be enabled.
4. Enter activation key.
   If you have a new activation key to be activated enter it in the empty edit box in the **Activation keys** section. Once you enter new activation key.
   - The **Add Another Key** button will be enabled, so that you can add a new edit box and enter another activation key.
5. Save License
   Once you have completed entering the activation key, you can press the **Save License** button for saving license files.

Once you press the **Save License** button

- It verifies the activation key.

- It creates a license file and stores it in the folder *LICENSEFOLDER\HMI Licenses\PANELID*.

Once the activation is completed, the enabled features or status of the activation keys will be displayed in the **Feature/status list** box that can be seen on the right side of the Activation keys boxes.

NOTE *After the activation is complete, to enable license in the panel you must restart it.*

37.3 Import license

If for some reasons it is not possible to have a working Internet connection on the PC running JMobile Studio, then you cannot use the previous two methods. You must download the license from a PC with an Internet connection and then import the license using the **Import License** button.

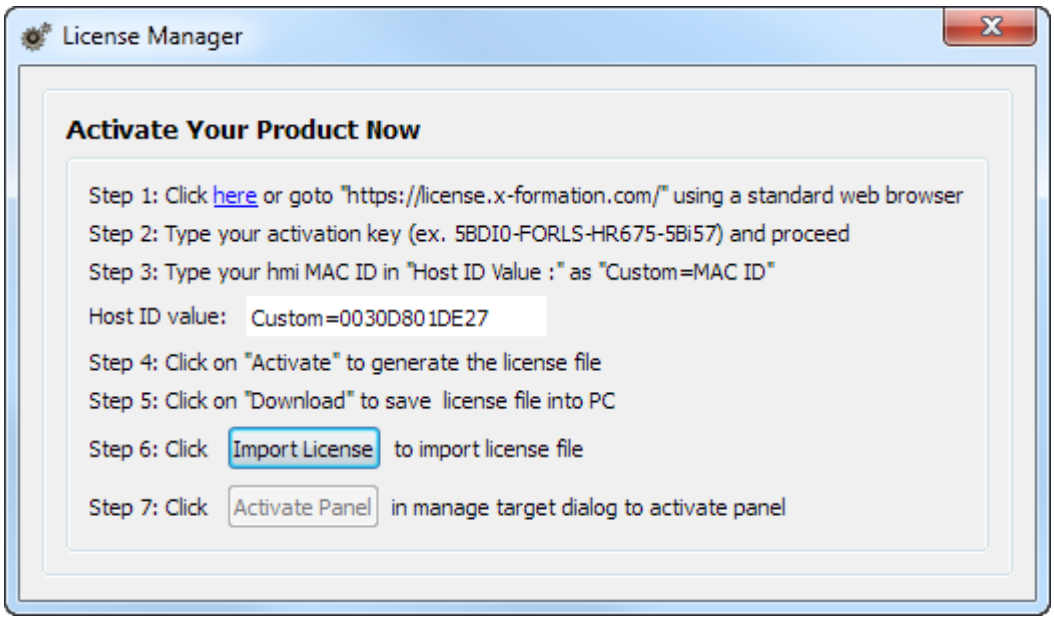

License Manager
Activate Your Product Now
Step 1: Click here or goto "https://license.x-formation.com/" using a standard web browser
Step 2: Type your activation key (ex. 5BDI0-FORLS-HR675-5Bi57) and proceed
Step 3: Type your hmi MAC ID in "Host ID Value :" as "Custom=MAC ID"
Host ID value: Custom=0030D801DE27
Step 4: Click on "Activate" to generate the license file
Step 5: Click on "Download" to save license file into PC
Step 6: Click Import License to import license file
Step 7: Click Activate Panel in manage target dialog to activate panel

Figure 312

1. Go to https://license.x-formation.com using a standard web browser.
2. Type your activation key (e.g. 5BDI0-FORLS-HR67G-5BI5T) and proceed.
3. Type your HMI MAC ID in field **HostID Value** as **Custom=MAC_ID** (e.g. Custom=0030D801DE27).
4. Click **Activate** to generate the license.
5. Click **Download** to save the license file to your PC.
6. Open JMobile Studio and select **Run > Manage Target > License** property page.
7. Select the panel if it is connected or enter the MAC ID if you are generating the license offline.
8. Click the **Import license** button and select the license obtained via step 5 above. Add more license files if more than one activation key was purchased.
9. You can press the **Activate Panel** button to activate licenses on the HMI Panel or you can press **Save License** for offline license file generation.

NOTE *After the activation is complete, to enable license in the panel you must restart it.*

38 Updating System Components in HMI Panels

Most of the system software components can be easily upgraded by the end users; this ensures a high degree of flexibility in providing updates and fixes to existing and running systems.

This upgrade can be done using USB flash drives, loaded with the new software modules, and by running the procedure, described in detail in this chapter.

Each unit comes from the manufacturing with a "product code" label, which includes all the information related to the factory settings (in terms of hardware, software and firmware components).

Product labeling is the first reference for checking the factory settings and version of the components installed at time of manufacturing.

The update tool on the HMI panel also provides the user with detailed information on the components actually running in the system.

NOTE *Files required for upgrades depend on the product code. Using the wrong files for upgrade may result in system malfunctions, and may even render the system unusable.*

NOTE *Files for upgrades are distributed on demand as a technical support activity.*

IMPORTANT The *downgrade of components is a very dangerous operation that could block machine and make it not more usable for HW/FW compatibility problems. Downgrade operations are not allowed and reserved to tech. support only.*

38.1 List of Upgradable Components

The HMI panels support the upgrade of the following components:

System Supervisor Firmware of the system supervisor controller
(sample file name: packaged_GekkoZigBee_v4.13.bin)

IMPORTANT *The System Supervisor Component can be upgraded only if the actual version on the panel is V4.13 or above. Version V4.08, V4.09, V4.10 and V4.11 MUST NOT be updated, they do not support automatic update from System Settings*

Main FPGA FPGA firmware
(sample file name: h146xaf02r06.bin)

Safe FPGA back-up copy of the Main FPGA that ensures unit booting in case of main FPGA corruption (may be after failed update)
(Sample file name: h146xaf02r06.bin)

NOTE *When updating FPGA firmware on the panel, the same file must be used for Main and Safe FPGA components*

Bootloader Loader to handle panel start-up
(sample file name: redboot_UN20HS010025.bin)

Main OS Main Operating System
(sample file name: mainos_UN20HS0160M0237.bin)

Configuration OS Back-up operating system that ensures units are recovering in case of main operating system corruption (may be after a failed update) (sample file name: configos_UN20HS0160C0237.bin)

38.2 Update of System Components from JMobile Studio

JMobile Studio provides a dialog to update system components by downloading them to the target device using the Ethernet communication interface.

The dialog is available in **Run** -> **Manage Target** -> **Board**.

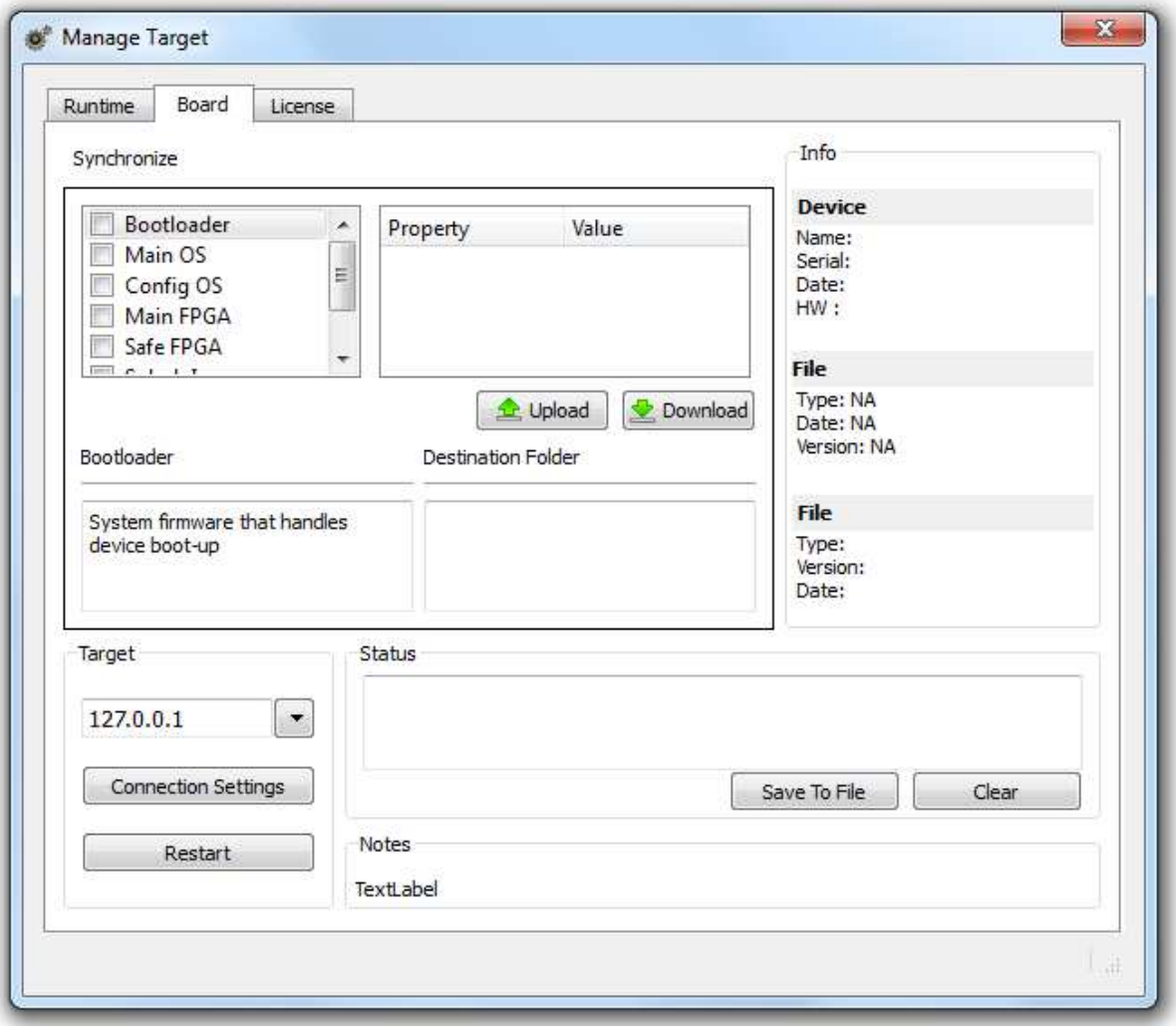


Figure 314

The first step is to use the Target discovery function to locate the panel IP from the local network. Click on the little arrow symbol and identify the HMI panel from the list of units recognized in the network. In case the panel is not listed, you can try a second time or type the IP directly in the box. Then click out of the box to accept the inserted IP. See the figure below.

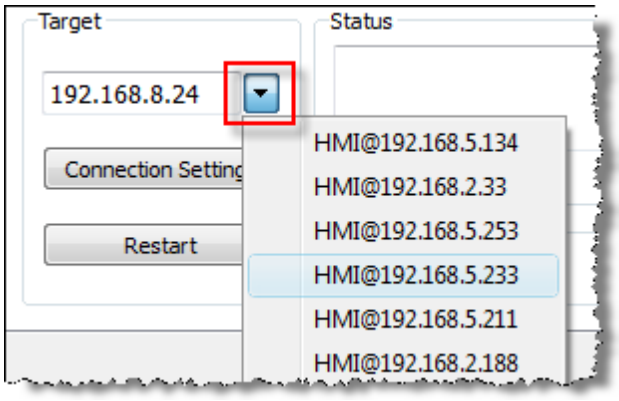


Figure 315

NOTE Discovery service is a broadcast service. When a remote connection is done via VPN or from external networks discovery is not working, so, type directly IP address of target to connect to it.

When the device is recognized the Info box shows the target details as shown as an example in the figure below.

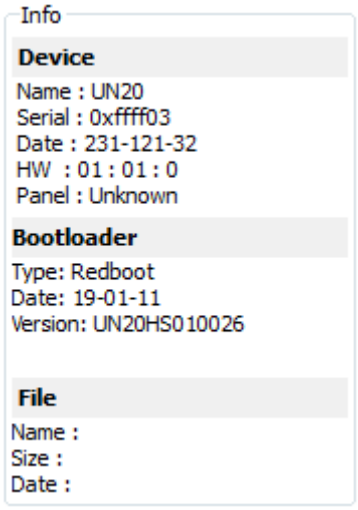


Figure 316

In the component list locate the one you need to update, check the box and browse for the file from the "Source file" box as shown in the figure below.

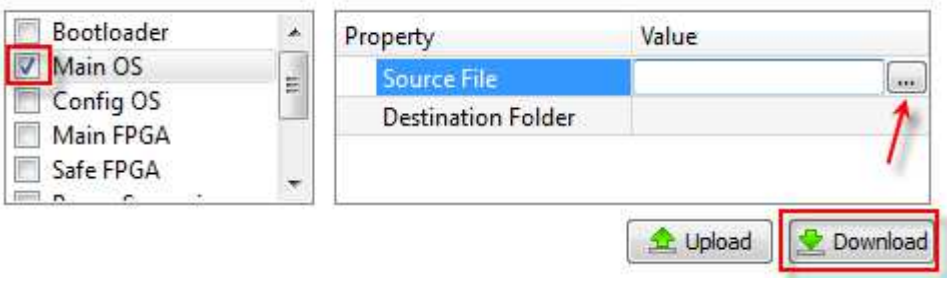


Figure 317

Then click download and check the progress from the Status box below.

NOTE *In the component selection you can mark more than one check box and provide the related file to be downloaded. The system will then execute the transfer of the all the elements, one after the other, and at the end you will need to cycle the power of the system.*

Manage target also allows you to replace the default splash screen image shown by the devices during the power up phase. Image for the splash screen must be provided in bitmap format saved in RGB 565 format.

NOTE *Splash screen images must NOT be bigger than 500 KB and they must have a black background to ensure the best optical results.*

38.3 Update of the System Components via USB Flash Drive

System components can be updated via USB flash drives. For each component a specific file is provided. Checksum file with an .md5 extension is required to be present in the same location as the system file to be upgraded.

To update a system component proceed as follow:

1. Copy all the files you need to upgrade to a USB Memory and plug this into the USB port of the panel.

2. Start the System Settings tool with the special procedure for getting this in **System Mode**, and then locate the desired item in the rotating menu.

3. Click directly on the item (the blue button with white label) and browse to locate the proper file stored on the pen drive (USBMemory). The figure below shows an example of the Main OS components.

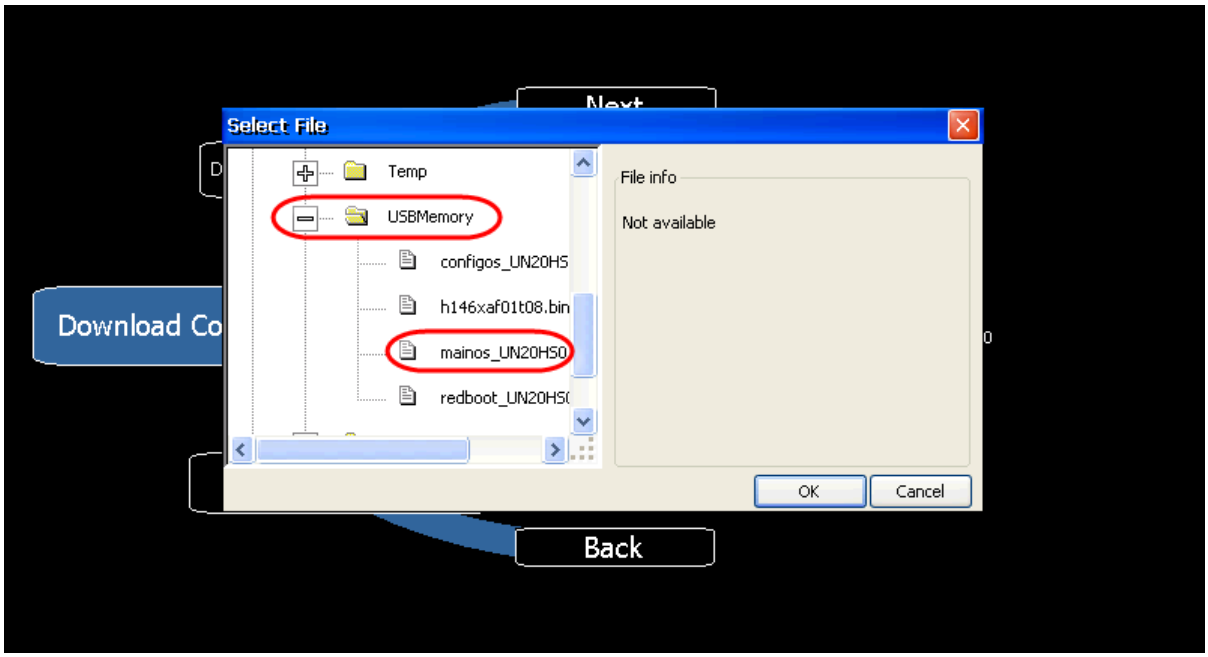


Figure 318

4. Select the "Download" command to transfer files to the panel. Select the "Upload" command to get files from the panel.

5. Follow the instructions on the screen to proceed with the update.

A progress bar on the screen will inform you about the status of the operation. Please make sure to NOT turn off power to the panel while a system component is being upgraded. Some of the components will require some time for the upgrade to complete.

NOTE *Upgrade procedure may change depending on the hardware revision or operating system version from which you start; please contact technical support offices for any detail about the exact sequence.*

39 Access Protection to HMI Devices

The following operations can be protected with a password set at device side:

- Runtime management: Install runtime and update runtime
- Board management: replace main BSP components such as MainOS, ConfigOS, Bootloader, etc
- Download and upload of project files

A default value for this password is used by the HMI Runtime and by JMobile Studio to access to device.

There are three ways to change device password in the HMI Runtime:

1) Using the tab **Remote** in the BSP Settings (in system mode) dialog box in System menu includes (starting from BSP versions V1.64 UN30/31 and V2.73 UN20).

Figure 319

2) Using the tab **Password** in Settings from the runtime Context Menu.

Figure 320

JMobile Studio shows a dialog asking for the password to match the password defined in the HMI device. The new password will be stored into the computer OS registry to be used for further connections.

3) Using **Set Target Password** in update package. Password is updated by the runtime just after the update process is completed. If the update failed (for example because **Old password** does not match the hmi password) a popup informs the user about it.

You can enter the same password in JMobile Studio using Manage Target -> Board -> **Connection setting** to allow in JMobile Studio to access runtime.

Default port used for this remote service is 2100.

NOTE *A format of hmi panel reset password device side.*
NOTE *For Win32 runtime, password is saved into Users\[username]\AppData\Roaming\Exor\buildNumber\server\config\RemoteUpdateConfig.xml.*
NOTE *Leave "Old password" empty as default if target password is not set.*

39.1 Ports & Firewalling

The table below shows all ports required by components of JMobile Suite:

Port	Usage	Remote Access	Board Management	Runtime/Project Management	Codesys iPLC
80/tcp	HTTP port	Yes		Yes	
21/tcp	FTP cmd port			Yes	
2100/tcp	Board port		Yes		
16384-17407/tcp	FTP data port (passive mode)		Yes	Yes	

990/udp	UDP broadcast (Device discovery)		Optional	Optional	
991/udp	UDP broadcast (Device discovery)		Optional	Optional	
998/udp	UDP broadcast (Device discovery)		Optional	Optional	
999/udp	UDP broadcast (Device discovery)		Optional	Optional	
5900/tcp	VNC Server	VNC only			
5100/tcp	JS Remote Debugger			Optional	
1200/tcp	Codesys 2.3 iPLC				Yes

Remote Access
Required If you need to connect to runtime using one of following:

- JMobile Client
- ActiveX
- Web access JM4Web

Runtime/Project Management ports
Required If you need to connect to runtime using in JMobile Studio for Runtime and Project operations like Update Runtime, Install Runtime, Download Project.

Board Management ports
Required if you need to connect to device with JMobile Studio for Board operations like Update BSP, download Splash Image etc.

NOTE *When broadcast service is not available (ex. in VPN networks), user have to type exact IP Address directly to reach device from JMobile Studio.*

40 Factory Restore

If you’re having problems with HMI device, you can try to restore factory default settings from **System Mode.**

To restore factory settings proceed as follow:

1) Enter in System Mode

2) Use one of the following operations available in rotating menu:
- Format Flash cleanup entire Flash disk and registry configuration.
- Restore Factory Settings allow user to select components to cleanup.

Both operations do not manage firmware factory restore (MainOS, ConfigOS, Bootloader, FPGA images, etc).

For more information related to Format Flash and Restore Factory Settings please ref. to System Mode

41 Tips and tricks to improve performance

JMobile Studio allow max flexibility for a projects designer. User can change svgs and replace images with customized versions. However, following some simple rules is possible to have faster projects in terms of boot time, page change and animations.

41.1 Static Optimization

Static optimization is a technique used in JMobile Studio to improve the runtime performance.

Using lot of graphics images and photos in a project might degrade performances. The idea of the static optimization is to merge many different images to a single background image in order to save rendering and loading times (only one raster image needs to be loaded and rendered instead of loading and rendering many single raster and/or vector images).

When you create a project in JMobile Studio, the pages might contain some widgets like texts, images, background images, background colors etc. Those widgets can be classified as

- **Static**: whose values or properties will not change at runtime (images and shapes, for instances)
- **Dynamic**: whose values or properties will change at runtime (for instance numeric fields and multistate images).

NOTE *Based on security settings, static parts of widgets could be not merged to background. This happens when a widget is configured as "hide" in security settings.*

IMPORTANT *When you change the properties of widgets with JavaScript you have to set the widget **Static Optimization** to **Dynamic**, otherwise changes to properties will be ignored.*

When downloading or validating a project, JMobile Studio identifies those static components and render them as background images to PNG-format files. These background images are saved as a part of the project under the folder called "/opt".

We can have:

- *full page background images*, containing all widgets that can be merged to page background
- *group background images*, containing all static widgets belonging to a group that can be merged together to form a group background (for instance the Gauge group is normally composed by a background a scale a label and a needle, where background scale and label can all be merged to a single background image)

The Static Optimization page attribute enables and disables static optimization of the whole page. If it is set to FALSE the optimization is disabled at all.

A finer control can be achieved using the static optimization attributes of each single widget:

- **Normal**: JMobile Studio automatically detects if the widget can be merged with the background. This is true if the widget is not a dynamic widget and does not overlap (i.e. it is stacked above) a dynamic widget.
- **Static**: The image is forced to be merged to background. This flag can be used when the static widget overlaps a dynamic transparent widget. In this case the automatic optimization will fail because it does not make any assumption on invisible areas (might be rendered at runtime).
- **Dynamic**: the widget is not optimized at all. For example we need to use this flag when a static widget needs to be changed by Javascript.

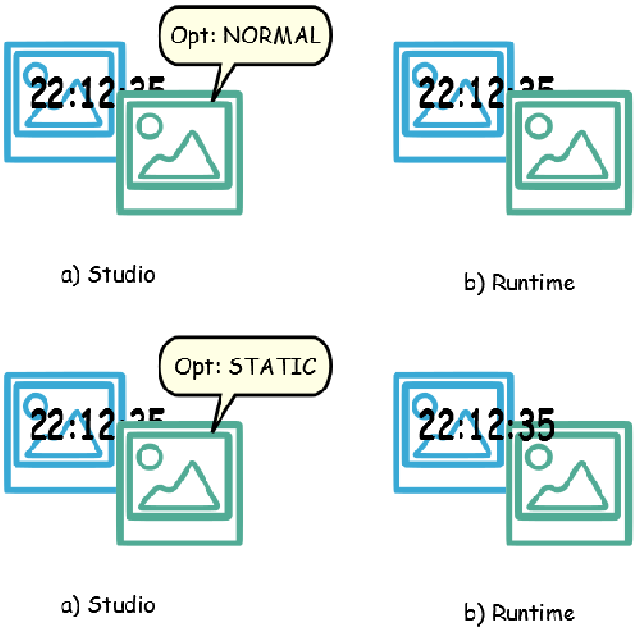


Figure 321

41.1.1 Best practices for max performance

1. Don't use static optimization at all if you have pages with almost only dynamic objects. In fact static optimization will save a lot of almost identical full size images for each page wasting a lot of memory that can be otherwise used to speed up project with other techniques (like for instance, page caching).
2. If possible avoid overlapping static widgets over a dynamic widget. This is the most important rule to follow. The overlapping area is computed considering the bounding rectangles of the widgets, that is the rectangles delimited by editing handles.
3. Bounding rectangles can include transparent areas. Try to minimize transparent areas (for example splitting the image in multiple images) since they can be a waste of resources even when optimized.
4. Optimize image size. The image will be rendered at a maximum size which is the size of the image widget containing the image. For best performances the widget needs to be the same size of the image.
5. If possible avoid using scale to fit which forces a rescaling at runtime for dynamic images and "hides" the actual image size during editing. In fact it is common to use very large images that are rescaled at runtime to fit their actual image widget size)
6. If overlapping cannot be avoided make sure to set the static widgets to back (order -> move to back). ie behind the dynamic widget (changing the z-order of static widgets to be under the dynamic widgets)
7. Use "size to fit" command to make the widget to the real size of his contents.
8. Choose the image file format based on the real target you have. JMobile Studio is supporting several raster formats like BMP, PNG, JPEG, TIFF and the vector format SVG:

	PROs	CONs
RASTER	- Fast rendering - Well standardized	- Big file size - Fixed resolution
VECTOR (SVG)	- Small file size - Rescale without quality loss - Can handle dynamic properties	- Complex SVG images with many graphic items and layers can be slow to render. - Scour software is free tool can be used to remove foreign code from file (http://www.codedread.com/scour/).

- Creating an optimized SVG is not simple.
- Only Tiny 1.2 (http://www.w3.org/TR/SVGTiny12/) supported.

9. Try to avoid using too many widgets in a page. Often widgets are placed outside of the visible area or their transparency is controlled by a tag. Since widgets are loaded even if they are not visible having too many widgets in a page can slow down significantly the page change time.
10. If possible, split a page with too many widgets in multiple pages with less widget.
11. For popping up new graphic elements in a page, prefer dialog pages with controlled positioning to transparent widgets
12. Have a check in opt folder to see if static optimization is working as expected: the widgets z-order might need to be changed to fix it
13. Numeric fields are often used to run JavaScript code on OnDataUpdate event even if the widget doesn't need to be visible on the page. In this case place the widget outside the page visible area instead of making it invisible, altering font color or visibility property. In fact in the latter case it is likely to end up with many left over wedges.
14. Use HotSpot button if you need a touch area to react to user inputs
15. If you reuse a widget from the gallery or you create your own, remember to set the right optimization properties (either static/dynamic/auto) or check if that kind of widget has the desired optimization properties. For example button widgets are dynamic widgets. For instance, if you use a button widget just for its frame it won't be optimized since the button widget is dynamic. If you just need the frame please use the UP Image SVG from the widget gallery.
16. With many pages having many dynamic widgets and using a common template the suggested configuration is to set:
    a. template static optimization flag set to **true**
    b. page static optimization flat set to **false** (background is already provided by the template)

    So, the background image can be reused among many different pages saving a lot of RAM that can be used for page caching.
17. Avoid using dynamic widgets like buttons only for the page background to make graphical effect without any real use of the buttons, instead use widget images (which can be obtained from the gallery) to apply the same graphical effect. This will reduce the load time of the page if we design it in an optimal way.

Follow an example of good and bad usage of Static optimization.

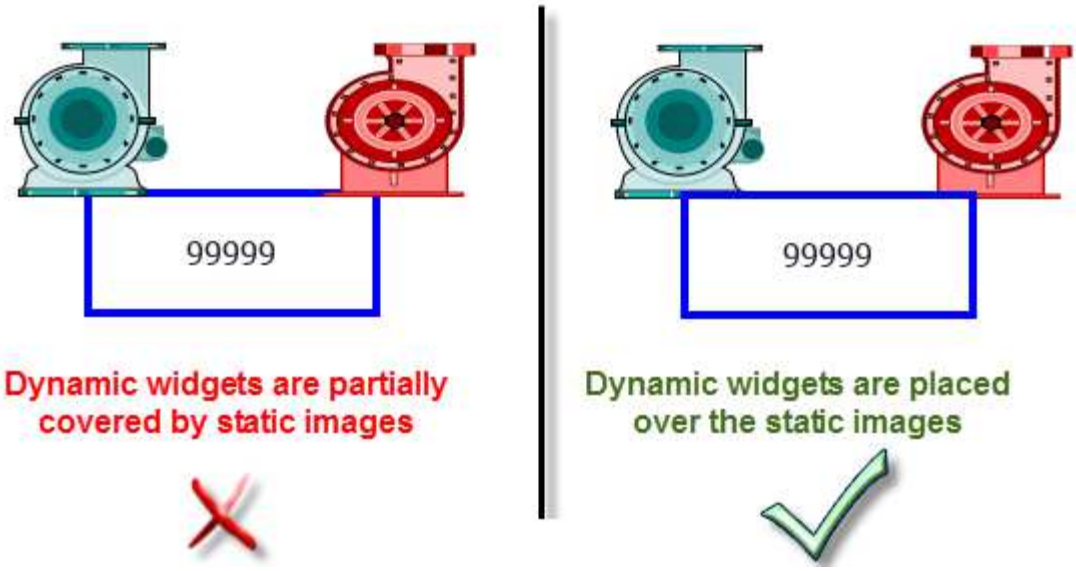


Figure 322

41.1.2 FAQ – Static optimization

Q: In a page where **there** are a **few identical** widgets, **in** the **OPT folder I see a png for each one of them. If they are really identical, why should** the **software duplicate them instead of having just 1 PNG?**

A: The software does not know if static images are actually the same since each widget could have different settings/properties altering the actual rendering at runtime

Q: Why are **the static images stored in a separate folder called Opt instead of storing** them **directly in the project folder?**

A: This solution avoids name collisions and allows skipping the upload of optimization images

Q: Why are **the static images stored as a *.png file instead of common *.jpg file?**

A: PNG format uses a lossless compression for images and support transparencies. JPEG files would render fuzzier compared to the PNG files with a different result in Studio (not using optimization) and runtime.

Q: What will happen when no optimization is done in the software?

A: Every single widget is rendered at runtime. In particular SVG images may require a lot of time to render in an embedded platform.

41.1.3 Templates

Currently, template pages can have large amount of static content. Still static optimization may not be applied to a template page, as it is decided based on the page where the template is used.
If a huge background image is repeated in every page via a template page, we tend to increase the footprint of the panel as the same static image is created for each of the page using the template page.

41.2 Page caching

Once accessed all pages are kept in a RAM cache up to the maximum allowed cache size depending on the actual platform's available RAM. This allows a much faster access since cached pages, once reloaded, only need to re-paint their content without reloading all page resources (images for instances).

41.3 Image DB

Image DB is a technique used to track usage of image files and amortize the cost of image loading by caching most frequently used images (for instance: Push Button images, Gauge needles, Slider thumbs etc.). So the same image used in many different places is loaded just once.

The image DB will preload the top most used images at start-up until memory limits are hit. This would further improve the individual page loading times.

The file imagecachelist.xml is created in project/opt folder, containing relevant information for ImageDB:

- Fill color. (in case of SVG)
- Size of the svg image
- Number of times an image is used in the project
- Number of different sizes for the same image

41.3.1 Best Practice to use the Image DB

1) Use uniform size of buttons, gauges and other widgets wherever possible.
2) Use same color themes as much as possible among widgets of same kind.

JMobile Studio provides three different levels of security:

1) **Project level: User Management**
   At project level you can define your Security Settings from JMobile Studio.
2) **Runtime level: Remote Password**
   At runtime level, you can protect upload and download of any project in your panel.
3) **OS level: BSP Password**
   At OS level, you can protect the update of the BSP components of your panel.

Below you can find an explanation on how to configure each level of security.

41.4 Project level

You can create your project Security Settings from JMobile Studio, by managing Users and User Groups.
The access to each object in the project will be given accordingly to User's settings.
Please refer to "*User Management and Password"* chapter for a detailed description of Security management at project level.

41.5 Runtime level

You can protect upload and download of any project in your panel by setting a dedicated password on the panel itself.
This password will also protect the remote Update Runtime procedure.
Please refer to the explanation below to learn how to set this password.

You can set the Remote Password by:

1) Context Menu
2) System Settings in System Mode
3) from JMobile Studio by means of an Update Package

Context Menu
Hold the finger on the screen to display the Context Menu, then select the "Settings..." voice.
It is then possible to set a new password or change an existing one from the "Password" tab of the Settings dialog.

System Settings in System Mode
Start the panel in System Settings / System Mode (using Tap Tap procedure).
Once accessed the rotating menu nevigate to the "BSP Settings" voice and select it.
It is then possible to set a new password from the "Remote" tab of the BSP info/settings dialog.
Note: available from BSP version 1.66 or higher.

JMobile Studio > Manage Target
From JMobile Studio select Run -> Manage Target
Click on the UpdatePackage button then, into the Update Package dialog, click on "Set Target Password" to set a new password or change an existing one.

41.6 OS level

41.7 Precache

Precache attribute of pages can be used by users to notify runtime to preload some pages in RAM at boot time for quicker access. Precache is useful for complex pages having many dynamic widgets.
In short, when precache is enabled on a page, access to the page is faster. As side effect precache slow down boottime because system will be not ready at boot until all pages with precache flag enabled has been cached in RAM.

41.7.1 Best Practice to use the Precache

1. Enable precache just for few pages having many dynamic widgets or pages frequently used by users.
2. Do not enable precache for all the pages because memory is not enough and risk is to have no benefit at all.
3. Disable static optimization for pages where precache is enabled can help to reduce memory used.

41.7.2 Frequently asked questions – Precache

How many pages should I enable for precache? Is there any limits?

Based on size & complexity of a page, the space required for precaching a single page can be in range 1/2Mb .. 3Mb (as mean).

Runtime proceed as follow when project is loaded:

1) Preload images of pages up to 76MB free (*imageDBLowMem*)
2) Preload pages with precahe=true up to 64MB free (*pageCacheLowMemMax*). In this phase images of these pages are loaded in RAM (into the Image DB).

When project is ready:

1) Any new page visited is saved in cache (RAM) with related images up to 40MB free (*pageCacheLowMemMin*)
2) When a page change happens and space in RAM is critical (<40MB), the runtime starts to cleanup cache (RAM) removing pages & related images up to 64MB free. In order, the runtime removes from cache:
    a. last visited pages and bigger and unused images (>320x240)
    b. if more memory is needed runtime can unload also pages in precache and all images loaded in Image DB.

42 FAQ

42.1 How to change fill color property according to Tag values

JMobile Studio allows to change the color property of a widget dynamically, basing on Tag values in two ways:

1) Using ColorPaletteCustom Xform
2) Connecting Color property to a String type Tag

42.1.1 Using ColorPaletteCustom Xform

1) Create a Tag (internal Tag or PLC Tag) that you want to refer to for the management of the color. Basing on the decimal value of this tag, the color will change accordingly. The tag can be of any data type.
2) Attach this Tag to the Fill Color property of an object (ex. a button).
3) Into the same dialog select now "XForms" tab and add a transformation by clicking on the [+] button. Select the "ColorPaletteCustom" transformation, then click near the "Palette" property:
4) Define now your custom palette by adding the colors that will be used for the object accordingly to the Tag value.
   The Index column reports the decimal value while the Color column shows the corresponding color.

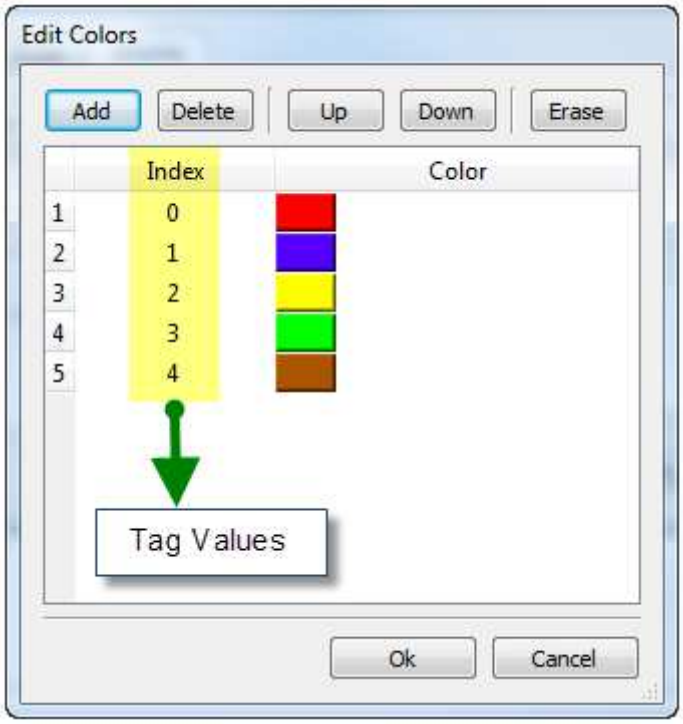


Figure 323

42.1.2 Connecting Color property to a String type Tag

1) Create a Tag (internal Tag or PLC Tag) that you want to refer to for the management of the color. Basing on the string content of this tag, the color will change accordingly.

The tag must be of String type and the Arraysize property of the tag (the string length) must be big enough to contain the string formatted as explained in the next steps.

2) Attach this Tag to the Fill Color property of an object (ex. button).
3) Now you can define the color writing inside the String Tag the RGB color code of the required color, you can use these formats:

#XXYYZZ
Where XX, YY and ZZ are the RGB components of the needed color expressed in Hexadecimal format, the range for these values is 00-FF

rgb(XXX,YYY,ZZZ)
Where XXX, YYY and ZZZ are the RGB components of the needed colors expressed in Decimal format, the range for these values is 0-255

Note: This feature can be applied to all the objects available into the Widget gallery that have a color property. The runtime change of the color is possible thanks to the properties of the SVGs that are composing the object, for this reason the color change can be applied on the objects that uses imported graphical files only if the graphical file is an SVG file created following the apposite guidelines, this feature can not be applied for example on jpeg or bmp files.

43 Functional Specifications and Compatibility

The scope of this chapter is to provide a clear overview of the supported functions and related limitations for both programming software and HMI Runtime system. What is listed below in this document is a safe limitation, above which proper operation and state-of-the-art performance of the system is not guaranteed.

43.1 Table of Functions and Limits

Function \ Feature	Max allowed
Number of pages	1000 (up to 10k x 10k as resolution based on HMI model)
Number of basic Widgets	2000 x page
Number of Tags	10000
Number of dialog pages	50 (max 5 can be opened in the same time)
Number of objects of any type in one page	2000
Number of Recipes	32
Number of parameter sets for a Recipe	1000
Number of elements per Recipe	1000
Number of user groups	50
Number of users	50
Number of concurrent remote clients	3 (ex. 1xJM4Web + 1xJMobile Client + 1x ActiveX)
Number of schedulers	30
Number of alarms	500/2000 (depends on HMI model)
Number of templates pages	50
Number of actions programmable per button state	32
Number of Trend Buffers	30
Number of curves per Trend Widget	5
Number of curves per page	10
Number of samples per Trend Buffer	200000
Number of Trend Buffer Samples for a Project	1200000
Number of messages in a message field	1024
Number of languages	12
Number of events per buffer	2048
Number of event buffers	4
JavaScript file size per page	16KB
Size of project on disk	30MB

43.2 Compatibility

Starting from the first official release of JMobile Studio V1.00 (00) we have applied the following policy for compatibility:

JMobile Studio version MUST always be aligned with JMobile Studio Runtime on the panel; the user has the responsibility to update Runtime components on the Target device together with any Studio update; a Runtime update can be done directly from Studio using the "Update Target" command available in the "Run\Manage Target" dialog.

Any version of Studio newer than V1.00 (00) is able to open and properly handle projects created on an older version, but no older than V1.00 (00).

Projects created with older versions of Studio, but not older than V1.00 (00), opened with later versions and deployed to compatible Runtime, are ensured to maintain the performance and functionality just as before.

Compatibility between newer versions of Runtime and those projects created and deployed with older versions of Studio is not ensured.

Do not edit projects with a version of JMobile Studio older than the one used to create them. It can result in a damage of the project and to runtime instability.

ptn0461-028.doc